INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR EM ENGENHARIA ELETRÔNICA

VICTOR LOMPA SCHWIDER

ACIONAMENTO DE COMANDOS POR VOZ APLICADO A UMA EMBARCAÇÃO MOVIDA A ENERGIA FOTOVOLTAICA

FLORIANÓPOLIS, 2025.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR EM ENGENHARIA ELETRÔNICA

VICTOR LOMPA SCHWIDER

ACIONAMENTO DE COMANDOS POR VOZ APLICADO A UMA EMBARCAÇÃO MOVIDA A ENERGIA FOTOVOLTAICA

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Eletrônica do Instituto Federal de Santa Catarina, para obtenção do título de Bacharel em Engenharia Eletrônica.

Área de concentração: Engenharia Eletrônica

Orientador: Prof. Dr. Flábio A. B. Batista

FLORIANÓPOLIS, 2025.

Ficha de identificação da obra elaborada pelo autor.

Schwider, Victor Lompa
Acionamento de comandos por voz aplicado a uma embarcação movida a energia fotovoltaica / Victor Lompa Schwider; orientação de Flábio Alberto Bardemaker Batista.
– Florianópolis, SC, 2026.
78 p.

Trabalho de Conclusão de Curso (TCC) – Instituto Federal de Santa Catarina, Câmpus Florianópolis. Bacharelado em Engenharia Eletrônica. Departamento Acadêmico de Eletrônica.
Inclui Referências.

1. Sistemas Embarcados. 2. Raspberry Pi. 3. Vosk. 4. Edge Impulse. 5. Downsample. I. Alberto Bardemaker Batista, Flábio. II. Instituto Federal de Santa Catarina. III. Acionamento de comandos por voz aplicado a uma embarcação movida a energia fotovoltaica.

ACIONAMENTO DE COMANDOS POR VOZ APLICADO A UMA EMBARCAÇÃO MOVIDA A ENERGIA FOTOVOLTAICA

VICTOR LOMPA SCHWIDER

Este trabalho foi julgado adequado para obtenção do título de Bacharel em Engenharia Eletrônica e aprovado na sua forma final pela banca examinadora do Curso Engenharia Eletrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 20 de fevereiro de 2026.

Banca Examinadora:

gov.br Documento assinado digitalmente
FLABIO ALBERTO BARDEMAKER BATISTA
Data: 21/02/2026 07:41:19-0300
Verifique em https://validar.iti.gov.br

Dr. Flábio A. B. Batista
Instituto Federal de Santa Catarina

gov.br Documento assinado digitalmente
HUGO MARCONDES
Data: 23/02/2026 13:51:03-0300
Verifique em https://validar.iti.gov.br

M.e Hugo Marcondes
Instituto Federal de Santa Catarina

gov.br Documento assinado digitalmente
RENAN AUGUSTO STARKE
Data: 23/02/2026 08:59:01-0300
Verifique em https://validar.iti.gov.br

Dr. Renan Augusto Starke
Instituto Federal de Santa Catarina

AGRADECIMENTOS

Primeiramente, minha eterna gratidão à minha família, Julio César Schwider, Suzana Rodrigues Lompa, Leonardo Lompa Schwider e Julia Lompa Schwider, pelo apoio incondicional, incentivo e principalmente pela paciência que tiveram durante toda essa caminhada que, apesar de tortuosa, sem dúvidas me trouxe muita alegria.

Agradeço também à minha namorada, Letícia Pereira da Luz, que esteve ao meu lado durante essa trajetória e, muitas vezes, mesmo sem saber, foi quem me fez continuar lutando por este sonho.

Ao meu orientador, Flábio Batista, e a todos os meus professores, agradeço pela orientação técnica e profissional, pela dedicação e pelo apoio acadêmico durante o desenvolvimento deste trabalho e ao longo de toda a minha formação, sempre com foco na excelência de ensino e um perceptível carinho por aquilo que fazem.

Aos colegas da equipe Zênite Solar, especialmente Gabriel Ayres, Ivan Junior Andreolla e Gustavo Vianna França, pelo acolhimento, colaboração e troca de conhecimentos, que foram fundamentais para a execução deste projeto, da concepção à entrega final.

Aos colegas de curso, hoje grandes amigos, que tornaram este percurso um prazeroso passeio, repleto de boas risadas, apoio, aprendizado mútuo e memórias que levarei com muito carinho para o resto da vida.

Por fim, a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho, deixo aqui o meu mais sincero agradecimento.

“As palavras têm poder. Elas podem criar realidades.”

— *Deepak Chopra*

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema embarcado utilizando a Raspberry Pi 3 B+ para controle por voz de uma embarcação movida a energia solar, com comunicação entre os módulos realizada por meio do protocolo *Controller Area Network* (CAN). O sistema foi projetado para funcionar de forma eficiente em ambientes com alto nível de ruído e sem conexão com a internet, utilizando o mecanismo de detecção de palavra de ativação e reconhecimento de fala *offline* com a biblioteca Vosk. A escolha dos componentes e tecnologias considerou a disponibilidade prévia dos materiais e a necessidade de baixo consumo energético e resposta rápida. São abordados os fundamentos de sistemas embarcados, comunicação CAN e tecnologias de reconhecimento de fala, com uma análise comparativa entre diferentes bibliotecas de reconhecimento automático de fala e diferentes algoritmos de reamostragem. Os resultados indicam que o Vosk oferece um bom equilíbrio entre acurácia, desempenho e viabilidade em sistemas de recursos limitados.

Palavras-Chaves: Sistemas embarcados, Raspberry Pi, protocolo CAN, reconhecimento de voz, Vosk, controle por voz *offline*, reamostragem.

ABSTRACT

This work presents the development of an embedded system using the Raspberry Pi 3 B+ for voice-controlled operation of a solar-powered boat, with communication between modules carried out via the Controller Area Network (CAN) protocol. The system is designed to operate efficiently in noisy environments and without internet connectivity. Also, the system is woken up by a wake word detection mechanism, starting the offline speech recognition model through the Vosk library. Component and technology choices were driven by material availability and the need for low energy consumption and quick response time. The theoretical framework includes embedded systems, CAN communication, and speech recognition technologies, along with a comparative analysis of different Automatic Speech Recognition libraries and different downsampling algorithms. Results indicate that Vosk offers a good balance between accuracy, performance, and feasibility in resource-constrained systems.

Keywords: Embedded systems, Raspberry Pi, CAN protocol, speech recognition, Vosk, offline voice control, downsample.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

DSB	Desafio Solar Brasil
STT	Speech-to-text
ML	Machine Learning
CAN	Controller Area Network
ASR	Automatic Speech Recognition
WER	Word error rate
ALSA	Advanced Linux Sound Architecture
CRC	Cyclic Redundancy Check
VAD	Voice Activity Detection
FFT	Fast Fourier Transform
MFCC	Mel-Frequency Cepstral Coefficients
ANC	Active Noise Cancellation
IA	Inteligência Artificial
MIC	Módulo de Interface do Controle
MCV	Módulo de Controle por Voz
EON	Edge Optimized Neural
SDK	Software Development Kit

SUMÁRIO

1. INTRODUÇÃO

A embarcação que constitui o objeto de estudo e o ambiente de aplicação deste protótipo, apresentada na Figura 1, é destinada a competições com provas de longa duração, como as realizadas no Desafio Barco Solar Brasil (DSB). O evento é composto por diferentes modalidades, incluindo provas de resistência (*endurance* ou raia longa), que podem ultrapassar quatro horas contínuas de navegação, além de provas de velocidade e manobrabilidade distribuídas ao longo de vários dias de competição. Nessas provas, o piloto permanece responsável pelo controle contínuo do motor e direção ao longo de extensos períodos. Em cenários desse tipo, a necessidade de interação constante com os sistemas pode causar fadiga física e mental, afetando tanto o desempenho do piloto quanto a segurança da operação. Diante desse contexto, surge a necessidade de soluções tecnológicas capazes de auxiliar o piloto, reduzindo sua carga de trabalho sem comprometer o controle do veículo.

Figura 1 - Guarapuvu 2

Fonte: Zênite Solar (2025)

Como forma não apenas de mitigar esse problema, mas também de viabilizar automações, controles mais elaborados da embarcação e interações mais pontuais entre o piloto e a embarcação, este trabalho propõe o desenvolvimento de um assistente por voz. A solução permite controlar o barco por meio da fala, possibilitando ao piloto manter o foco em tarefas como comunicação com a equipe em terra e monitoramento tanto do ambiente quanto do barco enquanto o sistema realiza o processamento e o encaminhamento das instruções aos módulos responsáveis pelo controle da embarcação.

1.1. Contextualização do projeto

O avanço das técnicas de inteligência artificial aplicadas ao reconhecimento de fala tem permitido a implementação de sistemas de interação por voz mesmo em plataformas com recursos computacionais limitados. Tecnologias de *speech-to-text* (STT), também conhecidas como transcrição de fala, e tecnologias de detecção de palavra de ativação (*wake word detection*) vêm sendo utilizadas em assistentes virtuais e dispositivos inteligentes, demonstrando potencial para aplicações em ambientes industriais, automotivos e náuticos.

No contexto de embarcações utilizadas em competições de eficiência energética, como as que participam do Desafio Barco Solar Brasil, a integração de sistemas inteligentes pode contribuir significativamente para a melhoria da experiência do piloto e para a evolução dos sistemas da embarcação. O Desafio Solar Brasil (DSB) é um projeto de extensão universitária que organiza uma competição de barcos movidos à energia solar visando estimular o desenvolvimento de tecnologias e aplicações das fontes alternativas de energia em embarcações, através do esporte e educação em tecnologia e meio ambiental. (DESAFIO SOLAR BRASIL, 2023).

Neste cenário, o projeto propõe a implementação do Módulo de Controle por Voz (MCV25) utilizando uma plataforma Raspberry Pi como unidade de processamento central. O sistema é capaz de detectar uma palavra de ativação, reconhecer comandos de voz e transmiti-los pela *Controller Area Network* (rede CAN), já utilizada no barco, ao Módulo de Interface do Controle (MIC19), responsável pelo controle do motor, da direção e do estado geral da embarcação. Dessa forma, o trabalho busca demonstrar a viabilidade técnica do uso de interfaces por voz em aplicações interativas voltadas ao controle veicular, contribuindo para o

desenvolvimento de soluções mais intuitivas, seguras e eficientes para interação homem-máquina em plataformas móveis.

Apesar da disponibilidade de soluções comerciais de reconhecimento de voz, como assistentes virtuais e módulos prontos, tais sistemas não se mostraram adequados ao contexto deste projeto. Em geral, essas soluções dependem de conexão contínua com a internet, possuem arquitetura fechada e não oferecem flexibilidade para integração direta com redes industriais, como a rede CAN. Além disso, apresentam limitações quanto à personalização de comandos, controle de latência e operação confiável em ambientes com ruído elevado. Por conta disso, optou-se pelo desenvolvimento de uma solução dedicada, embarcada e totalmente local, permitindo maior controle sobre o processamento, integração direta com os demais módulos da embarcação e adequação às exigências técnicas e operacionais do DSB.

1.2. Objetivo

Desenvolver e validar um assistente por voz embarcado capaz de reconhecer comandos de fala em língua portuguesa e transmiti-los, por meio de uma rede CAN, a um módulo de controle da embarcação, visando reduzir a carga de trabalho do piloto e ampliar as possibilidades de interação homem-máquina em uma embarcação movida a energia solar utilizada em competições de longa duração.

1.3. Objetivos específicos

- Implementar um sistema de reconhecimento de palavra de ativação otimizado para execução em plataforma embarcada de propósito geral.
- Desenvolver um módulo de transcrição de fala utilizando técnicas de inteligência artificial, com suporte à língua portuguesa.
- Realizar a captura e o processamento de áudio em tempo real em uma plataforma Raspberry Pi 3B+.
- Implementar a interpretação dos comandos de voz reconhecidos, convertendo-os em instruções estruturadas.

- Integrar o sistema de reconhecimento de voz a uma rede CAN, permitindo a comunicação com o módulo de controle da embarcação e demais subsistemas.
- Garantir que os comandos transmitidos via voz não interfiram no controle manual prioritário da embarcação.
- Avaliar o desempenho do sistema em termos de tempo de resposta, acurácia e confiabilidade em ambiente embarcado.
- Validar o funcionamento do assistente por voz em cenários representativos de operação da embarcação.

1.4. Resultados esperados

Espera-se que, ao final do desenvolvimento, o sistema proposto seja capaz de detectar de forma confiável a palavra de ativação e reconhecer comandos de voz do piloto de forma iterativa, mesmo em um ambiente sujeito a ruídos de vento e do próprio barco em operação. O assistente por voz deverá interpretar corretamente os comandos reconhecidos e transmiti-los via barramento CAN ao módulo MIC, permitindo o controle indireto de funções relacionadas ao barco.

Além disso, espera-se demonstrar a viabilidade técnica do uso de interfaces por voz em sistemas embarcados aplicados a competições de eficiência energética, evidenciando a ampliação das funcionalidades de controle do barco. Como resultado adicional, o projeto deverá fornecer uma base modular e extensível, possibilitando futuras expansões, como a inclusão de novos comandos, automações adicionais e integração com outros subsistemas da embarcação.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta os fundamentos teóricos que embasam o desenvolvimento do sistema proposto, cuja função é possibilitar a comunicação de módulos de uma embarcação movida a energia solar com uma interface de controle por voz, implementada em uma Raspberry Pi 3 B+. Os temas abordados abrangem sistemas embarcados, o protocolo de comunicação CAN, reconhecimento de voz *offline* e detecção de palavra de ativação.

2.1. Sistemas embarcados e plataformas de desenvolvimento

Sistemas embarcados são dispositivos computacionais projetados para realizar funções específicas em um sistema maior. Ao contrário de computadores de uso geral, os sistemas embarcados têm como principais características a eficiência, confiabilidade e, frequentemente, restrições quanto a consumo de energia, processamento e tamanho.

A utilização da plataforma Raspberry Pi 3 B+ para o desenvolvimento deste projeto se justifica por sua disponibilidade na equipe e por sua versatilidade. Por mais que já estivesse sendo utilizada com outro propósito, a plataforma disponível ainda possuía memória e CPU o suficiente para executar o projeto em paralelo.

A placa conta com um processador ARM Cortex-A53 quad-core de 1.4GHz, memória RAM de 1GB, conectividade sem fio (Wi-Fi e Bluetooth), além de diversas interfaces de comunicação, como GPIO, I2C, SPI e UART. Apesar de não ser a opção mais eficiente em termos de consumo energético, sua capacidade de processamento e compatibilidade com sistema Linux a tornam adequada para a execução de tarefas como reconhecimento de voz.

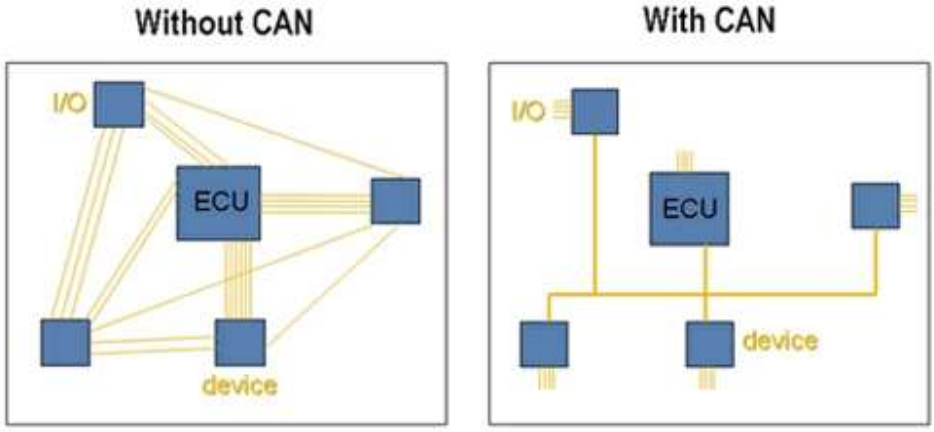
Embora a Raspberry Pi 3B+ possua recursos significativamente superiores aos de microcontroladores tradicionalmente utilizados em sistemas embarcados, como maior capacidade de memória e processamento, o reconhecimento automático de fala ainda impõe carga computacional considerável, especialmente durante a inicialização e execução contínua do modelo Vosk. Apesar disso, a implementação ideal do sistema seria por meio da utilização de uma placa própria para a aplicação.

2.2. Comunicação via Protocolo CAN

O *Controller Area Network* (CAN) é um protocolo de comunicação criado pela Bosch na década de 1980 para facilitar a troca de dados entre módulos eletrônicos em ambientes industriais e automotivos, onde a confiabilidade é crucial (BOSCH, 1991).

A rede CAN se destaca por oferecer uma solução de comunicação robusta, confiável e de baixo custo, muito utilizada em sistemas embarcados. Uma das suas principais vantagens é a redução de cabeamento, evidenciada na Figura 2, já que múltiplos dispositivos podem se comunicar por meio de um único barramento serial, diminuindo o peso e o custo total do sistema. Além disso, o modelo de comunicação por *broadcast* permite que todas as unidades conectadas recebam as mensagens transmitidas, cabendo a cada nó decidir se a informação é relevante facilitando a expansão e modificação da rede sem impactos significativos na arquitetura existente.

Figura 2 - Rede CAN reduz cabeamento



Fonte: National Instruments (2025)

2.2.1. Funcionamento do protocolo CAN

O protocolo CAN utiliza um barramento serial do tipo multi-mestre, no qual todos os nós conectados compartilham o meio de comunicação e possuem a capacidade de iniciar a transmissão de dados. O acesso ao barramento é gerenciado por um mecanismo de arbitragem baseado em prioridade, no qual a mensagem com identificador numérico mais baixo tem prioridade durante a transmissão. Esse processo de arbitragem é não destrutivo, garantindo que a transmissão da mensagem

de maior prioridade ocorra sem interferências, o que contribui para uma comunicação determinística e adequada a sistemas em tempo real.

Além disso, o protocolo incorpora mecanismos de verificação de erro por meio de código de redundância cíclica (CRC), nos quais cada quadro contém um campo de CRC utilizado para checagem de integridade. Quadros identificados como incorretos por qualquer nó são descartados, e sinais de erro podem ser gerados para notificar a rede sobre a falha detectada, tornando o sistema mais tolerante a falhas (NATIONAL INSTRUMENTS, 2025).

Na indústria automotiva, o barramento CAN é utilizado em sistemas de controle de motor, freios ABS, airbag, sistemas de entretenimento e controle de portas, devido à sua confiabilidade e à capacidade de operar em ambientes com altos níveis de interferência eletromagnética (BOSCH, 1991).

Da mesma forma, a indústria náutica adota o protocolo CAN em sistemas de controle e monitoramento de embarcações, como gestão de energia, propulsão elétrica, sensores ambientais, controle de leme e painéis de navegação. A padronização do protocolo CAN na náutica se reflete em protocolos como o NMEA 2000, uma derivação do CAN adaptada às necessidades de sistemas marítimos, suportando comunicação *plug-and-play* entre diferentes fabricantes.

2.3. ASR e suas abordagens

O *Automatic Speech Recognition* (ASR), ou reconhecimento automático de fala, é uma área complexa que combina conhecimentos de processamento de sinais, linguística, estatística e inteligência computacional. É o processo de converter áudio falado em texto, possibilitando a interação entre humanos e sistemas computacionais por meio da voz.

De acordo com Rabiner e Juang (1993), as técnicas de reconhecimento podem ser agrupadas em três grandes abordagens, cada uma com uma filosofia distinta sobre como modelar e interpretar o sinal de fala. Essa classificação nos dá uma perspectiva conceitual sobre os diferentes caminhos explorados na pesquisa de ASR, desde métodos baseados em características acústicas da fala até técnicas que utilizam modelos estatísticos e estratégias inspiradas em inteligência artificial.

2.3.1. A Abordagem Acústico-Fonética

A abordagem acústico-fonética parte da ideia de que a fala pode ser dividida em unidades fonéticas básicas, como fonemas, que possuem características acústicas próprias e que podem ser identificadas no sinal de áudio. Nessa abordagem, o sistema busca inicialmente detectar e classificar estas unidades, associando-as a sons da fala conhecidos, para depois combiná-las na formação de palavras e frases, com base em regras linguísticas que variam com o idioma.

Esse método assume que existem propriedades acústicas relativamente estáveis associadas a cada unidade fonética. Na prática, essas características variam bastante de acordo com o falante, o contexto e o ambiente, o que dificulta a segmentação e a classificação precisa dos fonemas. Por esse motivo, apesar de sua importância conceitual, essa abordagem apresentou limitações para aplicações práticas de maior escala (RABINER; JUANG, 1993).

2.3.2. A Abordagem por Reconhecimento de Padrões

Na abordagem de reconhecimento de padrões, o reconhecimento de fala é tratado como uma tarefa de comparação entre padrões do sinal de áudio e modelos previamente treinados. Diferentemente da abordagem acústico-fonética, não há necessidade de identificar explicitamente unidades fonéticas pois o sistema aprende, a partir de dados, como diferentes padrões acústicos se relacionam com palavras ou sequências de palavras.

Essa abordagem inclui técnicas baseadas em *templates* e, principalmente, modelos estatísticos, como os *Hidden Markov Models* (HMM), que se tornaram bastante utilizados em sistemas de ASR. O uso de modelos probabilísticos permite lidar melhor com as variações naturais da fala, como diferenças entre falantes e condições de gravação, tornando essa abordagem dominante em muitos sistemas de reconhecimento de fala ao longo de várias décadas (RABINER; JUANG, 1993).

2.3.3. A Abordagem de Inteligência Artificial

A abordagem de inteligência artificial busca incorporar estratégias mais flexíveis e adaptativas ao reconhecimento de fala, inspiradas em aspectos do

raciocínio humano. Nessa abordagem, o reconhecimento não depende apenas da comparação direta de padrões, mas também da capacidade do sistema de aprender relações complexas entre o sinal acústico e as unidades fonéticas, considerando diferentes níveis de informação.

Inicialmente, essa abordagem se baseou em sistemas nos quais o conhecimento sobre a fala e a linguagem era descrito manualmente, por meio de regras definidas por especialistas, que orientavam o processo de reconhecimento. Com o avanço das técnicas de aprendizado de máquina, passou-se a incluir modelos baseados em redes neurais artificiais, que permitem maior capacidade de generalização e adaptação. Segundo Rabiner e Juang (1993), essa abordagem representa uma evolução natural dos sistemas de ASR, ao integrar processamento acústico, aprendizado automático e informações linguísticas em um mesmo modelo.

2.4. Bibliotecas e frameworks de ASR

Sistemas de reconhecimento de fala *online*, como os utilizados por assistentes virtuais populares, dependem de conexão com servidores remotos para processamento. Em ambientes isolados ou com recursos de comunicação limitados, como é comum em embarcações, o reconhecimento *offline* é preferível.

Diversas bibliotecas e frameworks de reconhecimento automático de fala estão disponíveis para aplicações embarcadas, destacando-se soluções como o CMU Sphinx e o Vosk, utilizadas devido à capacidade de operação *offline* e à adequação para dispositivos com recursos computacionais limitados. O CMU Sphinx foi projetado com foco em flexibilidade e eficiência, oferecendo versões otimizadas para sistemas embarcados, como o PocketSphinx, que permite reconhecimento de fala em tempo real sem dependência de conexão com a internet (HUGGINS-DAINES et al., 2006). De forma semelhante, o Vosk fornece modelos acústicos compactos e compatíveis com arquiteturas ARM, possibilitando sua execução em dispositivos como a Raspberry Pi, com suporte a reconhecimento *offline* e baixa exigência de memória (ALPHA CEPHEI, 2020).

A Tabela 1 apresenta uma comparação entre diferentes soluções *open-source* de reconhecimento automático de fala com base em critérios técnicos relevantes para aplicações embarcadas. A métrica WER (*Word Error Rate*) representa a taxa de erro de palavras, indicando a proporção de palavras reconhecidas incorretamente pelo

sistema, onde valores menores indicam maior precisão. O suporte a leitura em tempo real refere-se à capacidade da solução de processar áudio de forma contínua e com baixa latência, característica essencial para aplicações interativas, enquanto a faixa de tamanho do modelo representa o intervalo aproximado de tamanho dos modelos, impactando diretamente o uso de memória e a viabilidade em dispositivos com recursos limitados. Requisitos de hardware descreve o tipo de hardware normalmente necessário para execução eficiente, como CPU ou GPU, e suporte a ajuste fino indica a possibilidade de adaptação dos modelos para domínios específicos ou vocabulários restritos.

Tabela 1 - Comparativo de soluções open source de STT

Solução	***Desempenho WER***	***Suporte a leitura em tempo real***	***Idiomas***	***Faixa de tamanho do modelo***	***Requisitos de hardware***	***Suporte a ajuste fino***
Whisper	10-30%	Apenas em lote (streaming via terceiros)	100+	39MB - 1.5GB	CPU/GPU Flexível	Limitado
Wav2Vec2	8-25%	Bom (requer adaptação para streaming)	50+	95MB - 300MB	GPU preferencial	Regular
Vosk	**12-35%**	**Bom**	**20+**	**50MB - 1.5GB**	**Eficiente em CPU**	**Limitado**
NeMo ASR	6-20%	Regular	15+	100MB - 1.1GB+	Obrigatório GPU	Regular
Speech Recognition	15-40%	Bom	10+	Varia conforme o backend	Apenas CPU	Nenhum
Coqui STT	13-30%	Bom	15+	50MB - 200MB	CPU/GPU Flexível	Regular
Mozilla DeepSpeech	15-35%	Limitado	15+	50MB - 200MB	CPU/GPU Flexível	Descontinuado
SpeechT5	9-25%	Limitado	10+	200MB - 600MB	Obrigatório GPU	Em pesquisa

Fonte: Adaptado de Foster (2025).

Modelos baseados em redes neurais profundas de grande escala, como o Whisper (RADFORD et al., 2022), apresentam elevada acurácia em tarefas de reconhecimento de fala, sendo disponibilizados em diferentes tamanhos de modelo, variando de 39 milhões a mais de 1,5 bilhão de parâmetros. No entanto, mesmo as

versões menores demandam capacidade computacional considerável, especialmente quando executadas exclusivamente em CPU.

O SpeechT5 é baseado em arquitetura *Transformer encoder-decoder*, a mesma em que se baseia a Whisper, e foi projetado para pré-treinamento em larga escala (AO et al., 2022). Considerando que modelos *Transformer* normalmente apresentam grande número de parâmetros e operações com custo computacional significativo, sua execução eficiente em plataformas embarcadas pode depender de técnicas adicionais de otimização.

O Mozilla DeepSpeech foi oficialmente descontinuado em 2021, deixando de receber manutenção ativa. Parte da comunidade deu continuidade ao projeto por meio do Coqui STT, mantido de forma independente. Ainda assim, a ausência de suporte institucional contínuo do DeepSpeech e a limitação evolutiva da arquitetura tornam sua utilização menos interessante.

O Coqui STT (COQUI, 2021), sendo de certa forma a continuação do DeepSpeech, preserva a mesma arquitetura baseada em redes neurais recorrentes com decodificação CTC. A execução padrão do framework ocorre por meio de ambiente Python, com interface principal disponibilizada via biblioteca e ferramentas em linha de comando. Além disso, o Coqui STT oferece suporte à execução offline e disponibiliza modelos pré-treinados, cuja execução em arquitetura ARM é possível mediante configuração adequada do ambiente.

Wav2Vec2 (BAEVSKI et al., 2020) e NeMo ASR (KOLUGURI et al., 2021) são arquiteturas modernas baseadas em redes neurais profundas de grande escala. Embora apresentem desempenho elevado em tarefas de reconhecimento de fala, seu tamanho e complexidade computacional podem tornar a execução direta em plataformas embarcadas com recursos limitados, como a Raspberry Pi 3 Model B+, desafiadora.

O Vosk (ALPHACEPHEI, 2022) é um framework de reconhecimento de fala baseado em modelos de redes neurais leves desenvolvido para execução eficiente em dispositivos de baixa potência, incluindo plataformas embarcadas com arquitetura ARM. O sistema oferece suporte a múltiplos idiomas, permite execução offline e disponibiliza interfaces em C++, Python e Java, facilitando a integração em aplicações embarcadas que exigem reconhecimento de comandos de voz em tempo real.

Além de suas características práticas, o Vosk se destaca por sua base arquitetural sólida construída sobre o toolkit Kaldi. Kaldi foi projetado como uma

ferramenta de pesquisa para ASR, fornecendo um conjunto de utilitários modulares e reutilizáveis para a construção de modelos acústicos e processos de decodificação (KALDI, 2009).

Para isso, o Vosk encapsula e otimiza as funções do Kaldi, criando uma versão mais leve e adequada para execução em dispositivos com recursos limitados, como o Raspberry Pi. Dessa forma, o Vosk mantém a precisão do Kaldi, mas com baixa latência e desempenho otimizado. Kaldi é uma das ferramentas mais utilizadas para desenvolvimento de sistemas de reconhecimento de fala e fornece o núcleo tecnológico sobre o qual o Vosk opera (ALPHACEPHEI, 2022).

Além disso, a base Kaldi permite personalizar gramáticas e modelos de linguagem, incluindo vocabulários específicos para contextos técnicos, como comandos náuticos, aumentando a acurácia em domínios restritos. Os modelos acústicos usados no Vosk são treinados com grandes conjuntos de dados de fala e comprimidos em formatos leves, otimizando a inferência para uso embarcado.

2.5. Detecção da palavra de ativação

A detecção da palavra de ativação (*wake word detection*) consiste na identificação contínua de uma palavra ou expressão específica em um fluxo de áudio seguida da extração de características acústicas relevantes. Essas características são então fornecidas para um modelo de Machine Learning (ML) para distinguir a palavra de ativação de outros sons e fala irrelevante. Quando a probabilidade de detecção ultrapassa um limiar pré-definido, o sistema considera que a palavra de ativação foi reconhecida e libera a execução das etapas subsequentes.

Essa abordagem é usada em sistemas embarcados e assistentes de voz para reduzir o consumo de energia e o processamento desnecessário, já que o reconhecimento completo de fala só é iniciado após a detecção da palavra de ativação. Em sistemas embarcados, a detecção de *wake word* é normalmente realizada de forma local, sem dependência de conexão com servidores remotos (EDGE IMPULSE, 2025). Essa estratégia aumenta a confiabilidade do sistema em ambientes com conectividade limitada e reduz a latência de resposta, características essenciais para aplicações interativas.

2.6. Abordagens de ASR nas ferramentas utilizadas

As abordagens clássicas para reconhecimento automático de fala descritas servem como base conceitual para entender o funcionamento de ferramentas modernas de reconhecimento de voz. Embora bibliotecas atuais não sigam rigidamente apenas uma das abordagens, é possível identificar elementos predominantes de cada uma delas em soluções como o Vosk e o Edge Impulse, utilizados neste projeto.

O Vosk é uma biblioteca de transcrição de fala que se baseia principalmente na abordagem de reconhecimento de padrões. O sistema utiliza características acústicas extraídas do sinal de áudio, como os *Mel-Frequency Cepstral Coefficients* (MFCC), e modelos estatísticos treinados a partir de grandes conjuntos de dados de fala (KALDI, 2009). Esses modelos aprendem a associar padrões acústicos a unidades fonéticas, sem a necessidade de identificação de fonemas por regras fixas, característica central da abordagem de reconhecimento de padrões. Além disso, o Vosk incorpora modelos de linguagem que auxiliam na escolha das sequências de palavras mais prováveis, aumentando o percentual de acerto do reconhecimento em fala contínua.

O Edge Impulse, utilizado neste projeto para a detecção da palavra de ativação, apresenta maior alinhamento com a abordagem de inteligência artificial. A ferramenta usa modelos baseados em aprendizado de máquina, como redes neurais, treinados diretamente a partir de dados coletados. Assim, o sistema aprende automaticamente a diferenciar padrões acústicos associados à palavra de ativação em relação a outros sons, sem a definição manual de regras fonéticas ou linguísticas. Essa capacidade de aprendizado a partir de exemplos reflete a evolução da abordagem de inteligência artificial.

2.7. Processamento de áudio em dispositivos embarcados

O processamento de áudio em dispositivos embarcados envolve a captura, filtragem, análise e interpretação de sinais sonoros utilizando recursos computacionais limitados. Esses sistemas são geralmente caracterizados por restrições quanto a energia, memória e capacidade de processamento, o que exige otimizações tanto no hardware quanto no software.

Em aplicações como o controle por voz em embarcações, o sistema precisa lidar com desafios adicionais, como a presença de ruído de fundo intenso do vento, da água e do motor. Além disso, são requisitos fundamentais a baixa latência para a aplicação interativa, o consumo energético reduzido e a capacidade de operação *offline*, uma vez que o sistema pode operar em locais com sinal de comunicação fraco ou inexistente.

Uma estratégia comum para lidar com as limitações de recursos em dispositivos embarcados é o *downsampling* do sinal de áudio, que consiste em reduzir a taxa de amostragem original. Essa técnica diminui a quantidade de dados a serem processados, reduzindo a carga computacional e o consumo de memória, ao mesmo tempo em que mantém informações relevantes para tarefas de reconhecimento de fala, desde que aplicada de forma adequada à faixa de frequência do sinal.

2.7.1. Downsampling

O *downsampling* pode ser também utilizado quando a frequência de aquisição do hardware de captura é superior à necessária para o processamento subsequente. De acordo com a teoria de amostragem, o processo de *downsampling* deve ser precedido por uma filtragem passa-baixa (*anti-aliasing*) para remover componentes espectrais acima da nova frequência de Nyquist, evitando o fenômeno de *aliasing*, no qual componentes de alta frequência são refletidas para regiões mais baixas do espectro, distorcendo o sinal digital (OPPENHEIM; SCHAFER, 2010).

Em sistemas embarcados, a implementação de filtros ideais pode ser inviável, motivando o uso de métodos simplificados de reamostragem. Entre esses métodos destaca-se o *nearest neighbor*, que seleciona diretamente amostras do sinal original, apresentando baixo custo computacional, porém sem qualquer filtragem *anti-aliasing*. A interpolação linear estima novos valores a partir das amostras adjacentes, resultando em uma forma de onda mais suave, mas igualmente limitada para *downsampling*, uma vez que não atenua adequadamente componentes de alta frequência (OPPENHEIM; SCHAFER, 2010).

Como alternativa, a média por blocos (*box averaging*) consiste na obtenção de cada amostra reamostrada a partir da média de um conjunto de amostras do sinal original, sendo equivalente à aplicação de um filtro passa-baixa simples do tipo média móvel seguido de decimação. A decimação corresponde ao processo de redução da

taxa de amostragem por um fator inteiro, no qual apenas uma em cada M amostras é mantida após a filtragem, reduzindo a quantidade de dados representativos do sinal (OPPENHEIM; SCHAFER, 2010).

Esse método promove uma atenuação parcial de altas frequências com baixo custo computacional, sendo descrito na literatura como uma solução prática para sistemas com recursos limitados. No contexto do reconhecimento automático de fala, métodos simplificados de *downsampling* tendem a ser suficientes, uma vez que técnicas de extração de características, como MFCC, são relativamente resistentes a pequenas distorções no sinal de entrada (RABINER; JUANG, 1993).

2.7.2. Pré-processamento

Para reduzir a influência do ruído ambiente, a literatura descreve diversas técnicas de pré-processamento de áudio, como algoritmos de supressão de ruído (LOIZOU, 2013), detecção de atividade de voz (*Voice Activity Detection*) (RABINER; JUANG, 1993) e técnicas de *beamforming* (BRANDSTEIN; WARD, 2001) em sistemas com múltiplos microfones. Essas abordagens têm como objetivo melhorar a relação sinal-ruído e restringir o processamento aos trechos relevantes do sinal de fala.

No contexto deste projeto, essas técnicas não foram empregadas. O microfone utilizado já possui cancelamento ativo de ruído, o *Active Noise Cancellation* (ANC), e tanto o Edge Impulse quanto a biblioteca Vosk realizam etapas de pré-processamento e extração de características que conferem mais resiliência ao sistema para níveis moderados de ruído. Além disso, a adição dessas técnicas adicionais implicaria em maior consumo computacional e aumento da complexidade do sistema, sem apresentar benefícios significativos para a aplicação proposta.

2.7.3. Conversão e processamento do áudio

No reconhecimento automático de fala, o sinal de áudio captado pelo microfone não é processado diretamente no domínio do tempo, pois essa representação apresenta alta sensibilidade a ruídos e variações de amplitude, além de não representar de forma adequada aspectos importantes da fala, como as frequências predominantes, a forma do espectro sonoro e os padrões que diferenciam os sons das palavras (RABINER; JUANG, 1993). Dessa forma, o sinal é inicialmente

convertido para o domínio da frequência por meio da Transformada Rápida de Fourier (FFT), aplicada em janelas curtas do sinal, o que permite identificar a distribuição de energia nas diferentes faixas de frequência ao longo do tempo. A partir do espectro obtido pela FFT, são extraídas características acústicas mais compactas e relevantes, sendo os coeficientes cepstrais na escala Mel (MFCC) uma das abordagens mais utilizadas.

Os coeficientes cepstrais na escala Mel (MFCC) são utilizados em sistemas de reconhecimento automático de fala devido ao seu desempenho superior na representação do sinal de voz, conforme demonstrado por Davis e Mermelstein (1980). Posteriormente, Rabiner e Juang (1993) destacam que representações cepstrais baseadas na escala perceptual Mel tornaram-se abordagem consolidada em sistemas de reconhecimento de fala, por conta de sua eficiência na modelagem das características espectrais relevantes da fala humana.

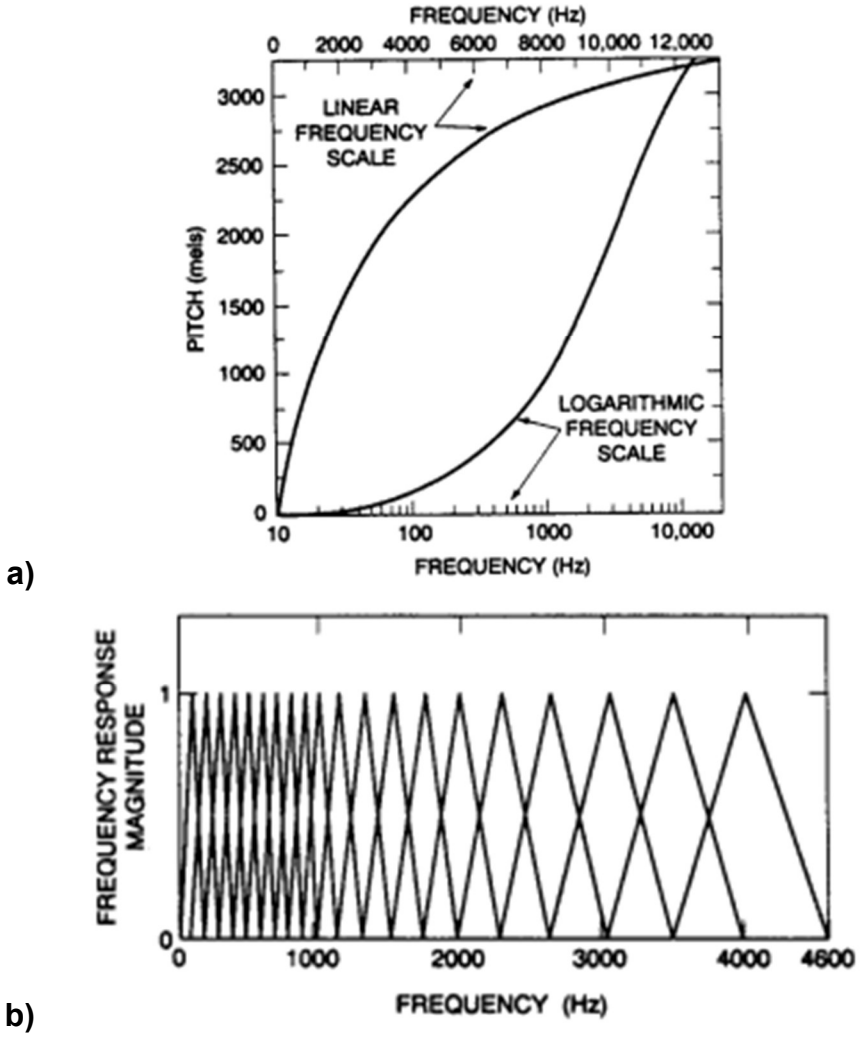
A escala Mel, apresentada na Figura 4a, representa a relação entre a frequência real em Hertz (Hz) e a percepção de altura sonora (*pitch*) pelo sistema auditivo humano. Observa-se que essa relação não é linear: para frequências baixas, pequenas variações em Hertz correspondem a variações perceptíveis significativas, enquanto em frequências mais altas são necessárias variações maiores em Hertz para que a diferença seja percebida. A curva ilustrada na figura evidencia esse comportamento não linear, aproximando-se de uma relação logarítmica em altas frequências. Essa característica justifica a utilização da escala Mel em reconhecimento de fala, pois permite maior resolução espectral nas faixas onde se concentram as principais informações fonéticas da fala humana.

A Figura 3b apresenta um exemplo de banco de filtros triangulares distribuídos ao longo do espectro segundo a escala Mel. Nota-se que os filtros são mais estreitos e mais densamente espaçados nas baixas frequências, enquanto se tornam progressivamente mais largos e espaçados nas altas frequências. Cada filtro calcula a energia presente em sua respectiva faixa espectral, resultando em um conjunto de valores que representam a distribuição de energia do sinal segundo a escala perceptual Mel. Essa distribuição não uniforme reflete diretamente o comportamento perceptual mostrado na Figura 3a, concentrando maior detalhamento onde o ouvido humano é mais sensível.

No processo de extração dos MFCC, o espectro obtido pela FFT é passado por esse banco de filtros, resultando em um conjunto de valores de energia na escala Mel.

Em seguida, aplica-se o logaritmo dessas energias e uma Transformada Cosseno Discreta (DCT), gerando coeficientes que representam de forma compacta a envoltória espectral da fala (DAVIS; MERMELSTEIN, 1980). Essa representação reduz dimensionalidade, mantém as informações mais relevantes para discriminação fonética e melhora a robustez frente a ruídos e variações de amplitude.

Figura 3 - Escala Mel



Fonte: RABINER, JUANG (1993)

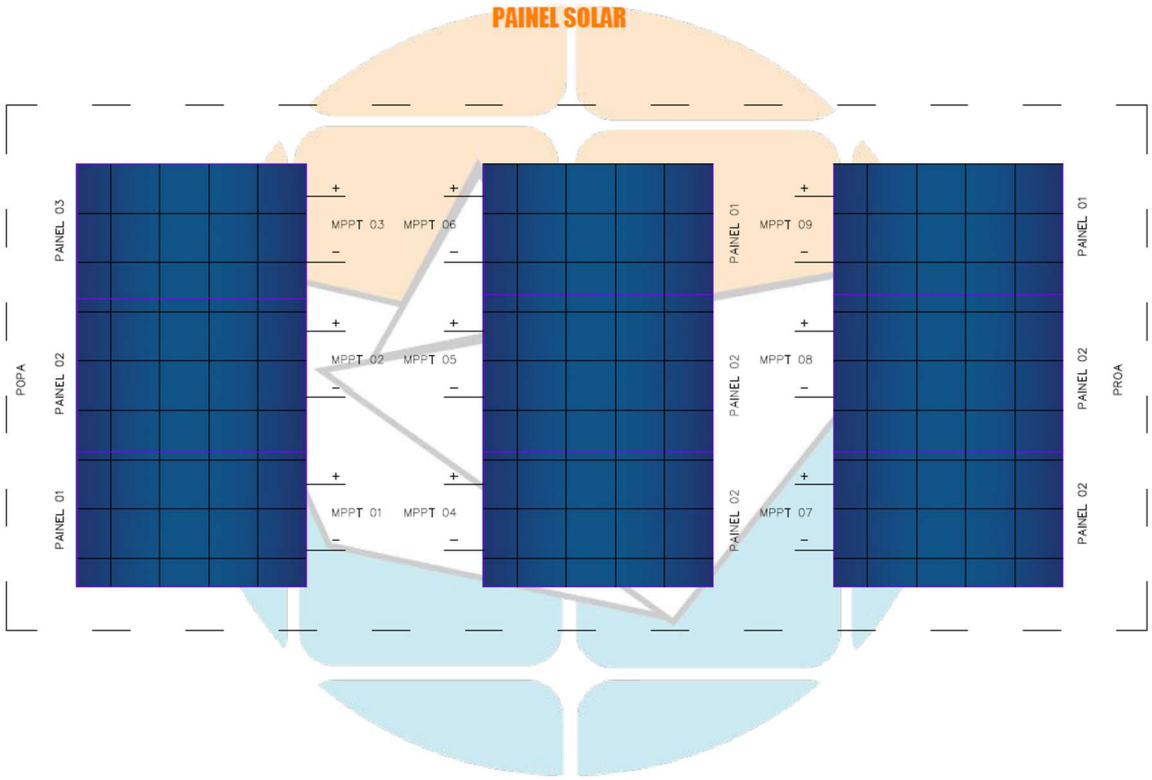
As ferramentas e bibliotecas utilizadas neste projeto, como o Edge Impulse para a detecção da palavra de ativação e o Vosk para reconhecimento de fala, utilizam representações espectrais baseadas na FFT e na extração de MFCC como entrada para seus modelos de aprendizado de máquina, viabilizando a execução *offline* e em tempo real em plataformas com recursos limitados.

2.8. Arquitetura do barco

O sistema elétrico da embarcação é organizado de forma modular, distribuindo energia proveniente dos painéis solares para os diversos subsistemas operacionais. A arquitetura é projetada para maximizar a eficiência energética e garantir autonomia em ambiente embarcado, com ênfase em confiabilidade e segurança.

Conforme apresentado na Figura 4, a energia elétrica é inicialmente gerada pelos painéis solares fotovoltaicos, organizados em múltiplos *MPPTs* (*Maximum Power Point Tracking*, ou Rastreamento do Ponto de Máxima Potência) que otimiza a captação de energia em diferentes níveis de incidência de raios solares. A energia captada é direcionada para o banco de baterias principais, que armazenam a energia gerada e fornecem tensão contínua aos demais módulos do barco.

Figura 4 - Painéis Solares

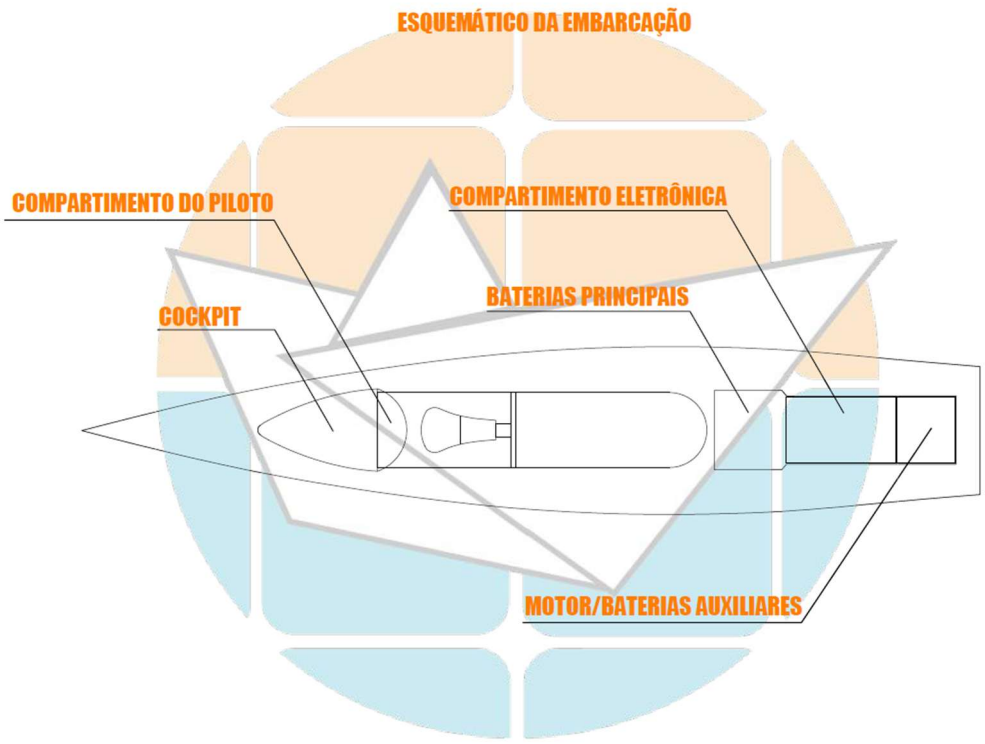


Fonte: Zênite Solar (2025)

A Figura 5 apresenta o esquemático da embarcação, com uma vista superior que ilustra a disposição física dos principais módulos e subsistemas a bordo. Neste diagrama, é possível identificar diversos elementos importantes:

- Cockpit: compartimento onde se encontra a eletrônica responsável pela interface homem-máquina, contendo sistemas auxiliares e de controle da embarcação.
- Compartimento do Piloto: área de operação e controle do piloto, onde estão concentradas as interfaces de comando, volante, botões de emergência e chaves.
- Compartimento da Eletrônica: contém os *MPPTs*, módulos de controle, contatoras, módulos de carregamento e interfaces de monitoramento do sistema elétrico.
- Baterias Principais: responsáveis pelo armazenamento da energia elétrica proveniente dos painéis solares e pelo fornecimento de tensão contínua para os sistemas críticos da embarcação.
- Motor e Baterias Auxiliares: localizados na parte traseira da embarcação, fornecem potência para propulsão e suportam cargas secundárias ou de backup.

Figura 5 - Esquemático da embarcação

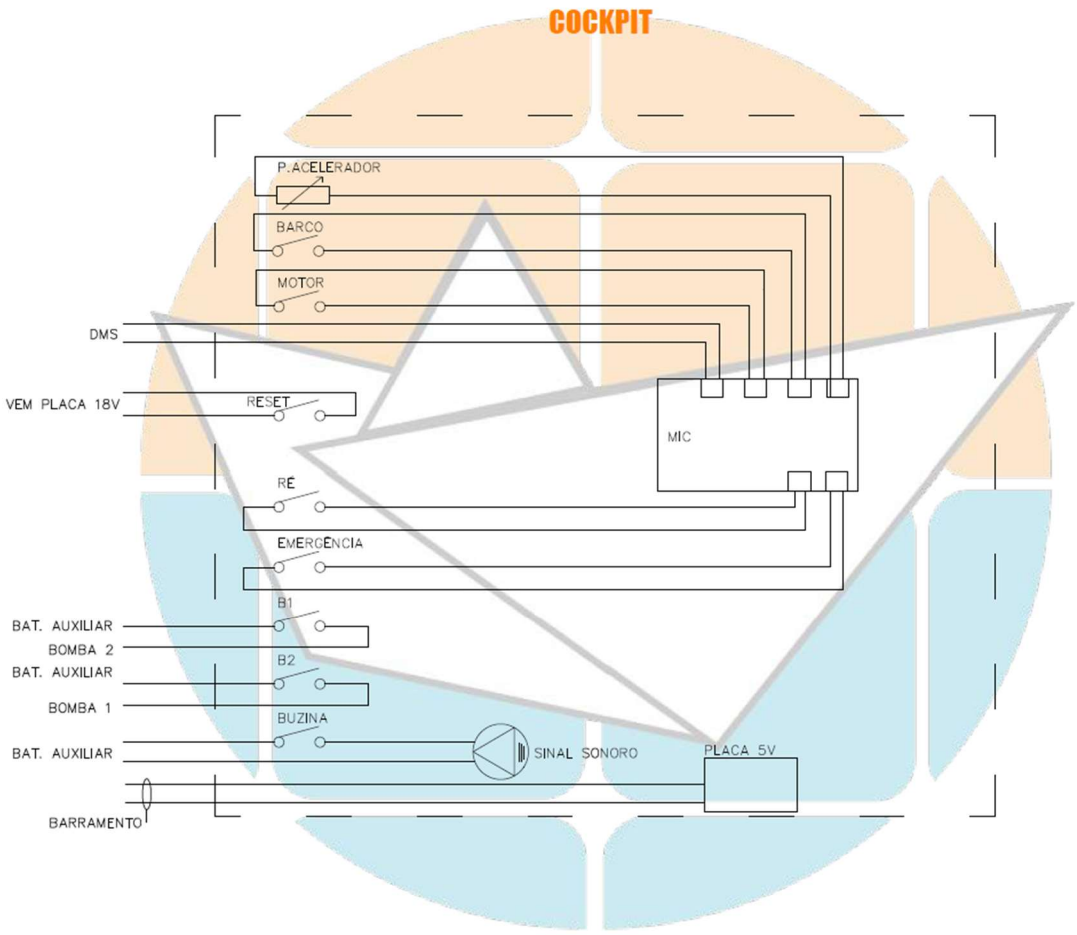


Fonte: Zênite Solar (2025)

A Figura 6 apresenta o diagrama elétrico do cockpit da embarcação, destacando a integração do módulo MIC19 com os demais sistemas. O MIC19 é responsável por diversas funções críticas da embarcação, incluindo o acionamento

das bombas de drenagem de água, a conversão da rotação do volante em variações no ângulo da rabeta, o controle da velocidade do motor, realizado a partir da leitura de um potenciômetro e a verificação do estado do *dead man switch*, cuja finalidade é identificar situações em que o piloto possa ter caído ao mar.

Figura 6 – Esquema elétrico do cockpit



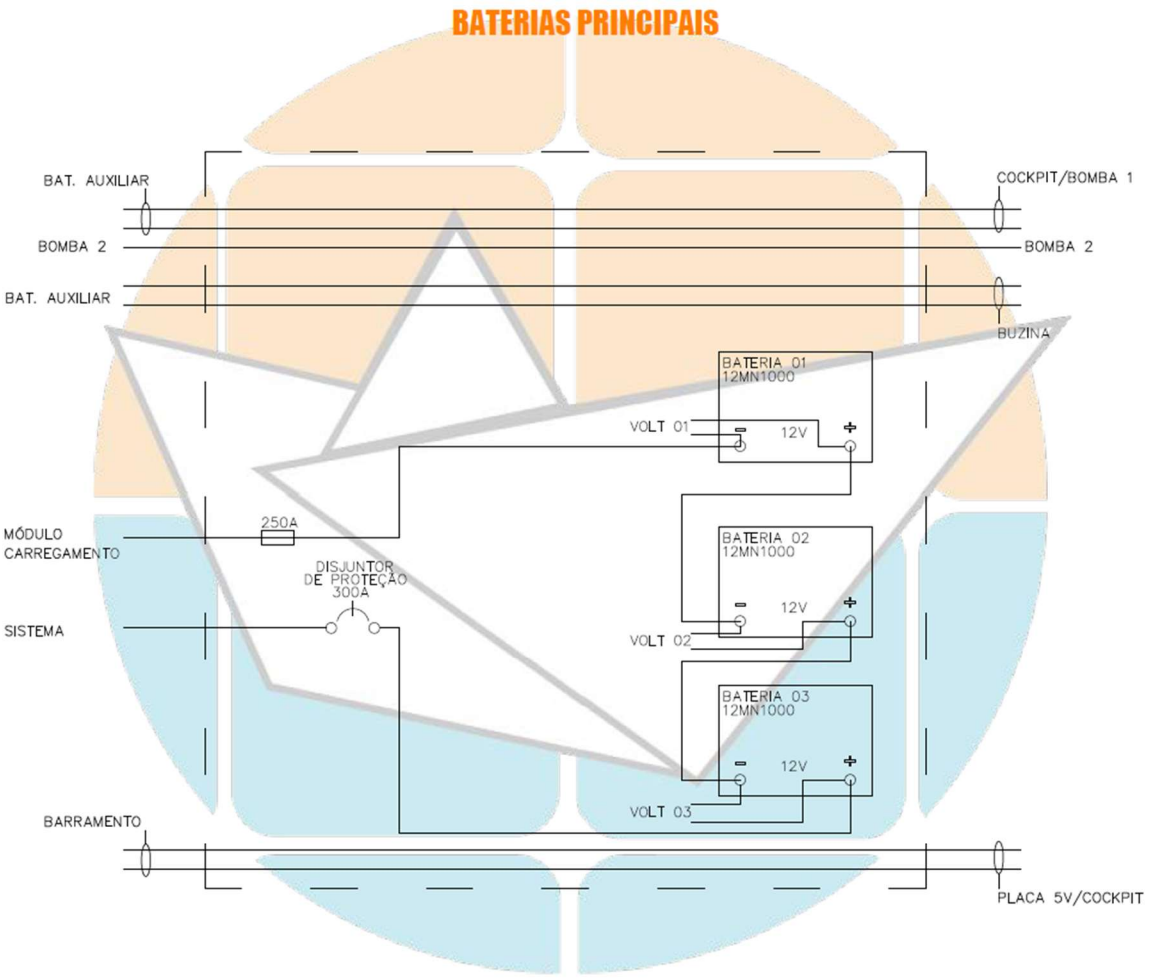
Fonte: Zênite Solar (2025)

O compartimento do piloto é o espaço central de operação da embarcação, reunindo todos os elementos essenciais para o controle e monitoramento do barco. Nele estão presentes botões e chaves de comando, buzina, volante, DMS, rádio e interfaces diversas, garantindo que o piloto tenha acesso rápido e seguro a todas as funções necessárias para navegação e operação da embarcação.

O esquema elétrico das baterias principais da embarcação, apresentado na Figura 7, é composto por três baterias de 12V conectadas em série, garantindo a tensão necessária para alimentar os sistemas do barco. Cada bateria possui

monitoramento individual de tensão, permitindo que os valores sejam exibidos diretamente no painel para acompanhamento do piloto.

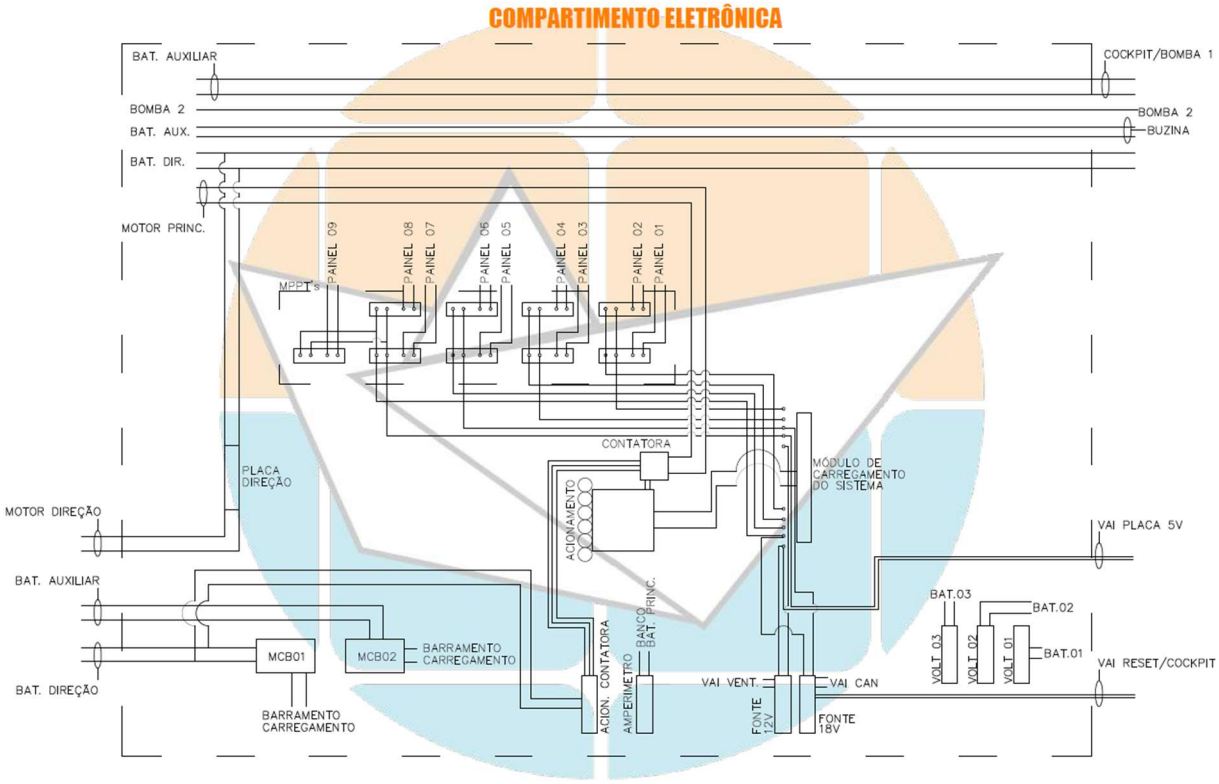
Figura 7 – Esquema elétrico das baterias principais



Fonte: Zênite Solar (2025)

O esquema elétrico do compartimento da eletrônica, apresentado na Figura 8, detalha a ligação dos painéis solares com os *MPPTs* e com o módulo de carregamento do sistema. No compartimento também estão presentes diversos módulos de conversão de energia, incluindo uma placa que converte a tensão para 18 V, necessária para alimentar a rede CAN, garantindo a comunicação entre os módulos da embarcação. Além disso, o compartimento abriga a contatora, responsáveis pelo acionamento seguro dos sistemas de potência, a placa de direção, e outros componentes eletrônicos essenciais para o funcionamento integrado do barco, como sensores e interfaces de monitoramento. Essa disposição garante que a energia seja distribuída de maneira eficiente, segura e organizada para todos os subsistemas.

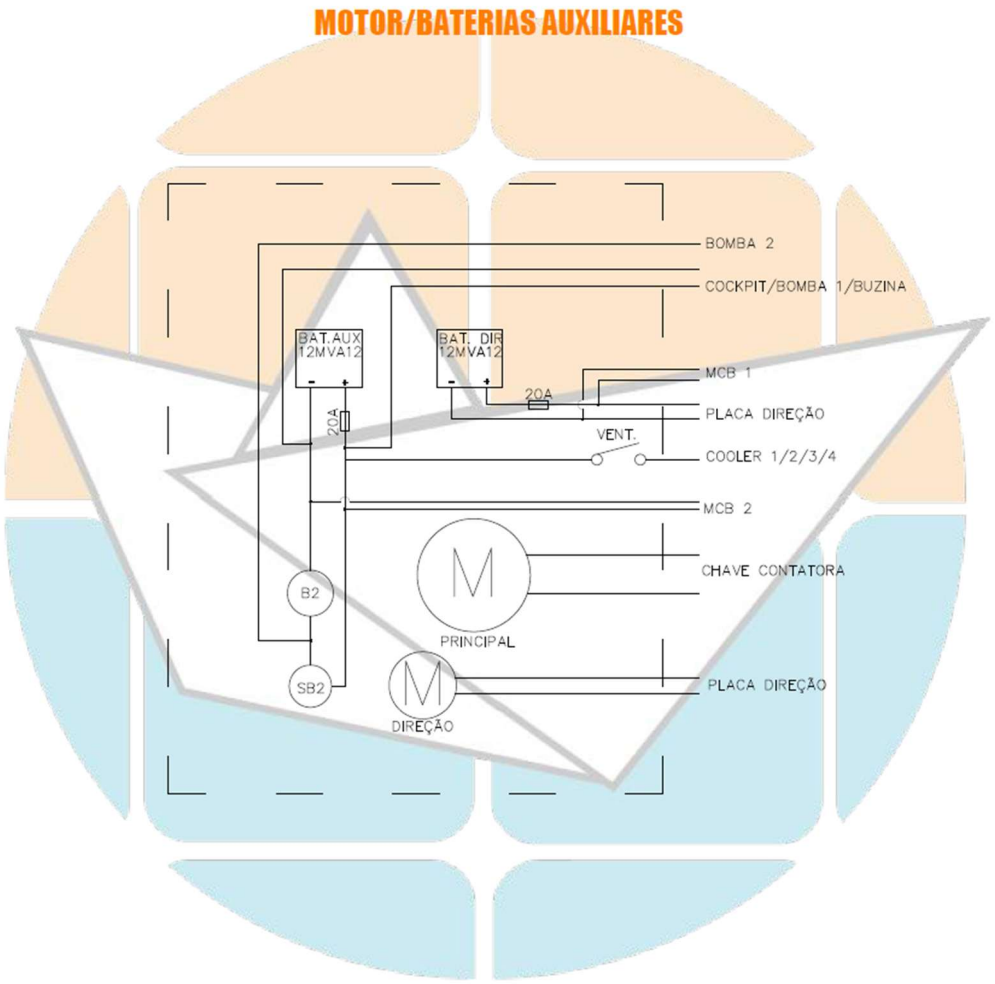
Figura 8 – Esquema elétrico do compartimento da eletrônica



Fonte: Zênite Solar (2025)

Por fim, a Figura 9 apresenta o esquema das baterias auxiliares e do motor da embarcação. O sistema é composto por duas baterias auxiliares, utilizadas para alimentar subsistemas específicos e independentes do banco principal de energia. A primeira bateria auxiliar é responsável pelo fornecimento de energia para as bombas de drenagem e para a buzina, garantindo o funcionamento contínuo desses dispositivos, considerados essenciais para a segurança da embarcação. A segunda bateria auxiliar é dedicada exclusivamente ao sistema de direção, fornecendo alimentação independente para a rabeta. Essa separação de fontes de energia atende às exigências estabelecidas pelo regulamento do Desafio Solar Brasil (DSB) e contribui para o aumento da confiabilidade operacional e da segurança durante a navegação.

Figura 9 – Esquema elétrico do compartimento do motor e baterias auxiliares



Fonte: Zênite Solar (2025)

3. METODOLOGIA

A metodologia adotada neste trabalho baseia-se no desenvolvimento incremental de um protótipo funcional de assistente por voz embarcado, integrando técnicas de reconhecimento de fala, processamento de áudio em tempo real e comunicação por barramento CAN. O projeto foi conduzido de forma experimental e aplicada, com foco na validação prática da solução em uma plataforma embarcada de recursos computacionais limitados.

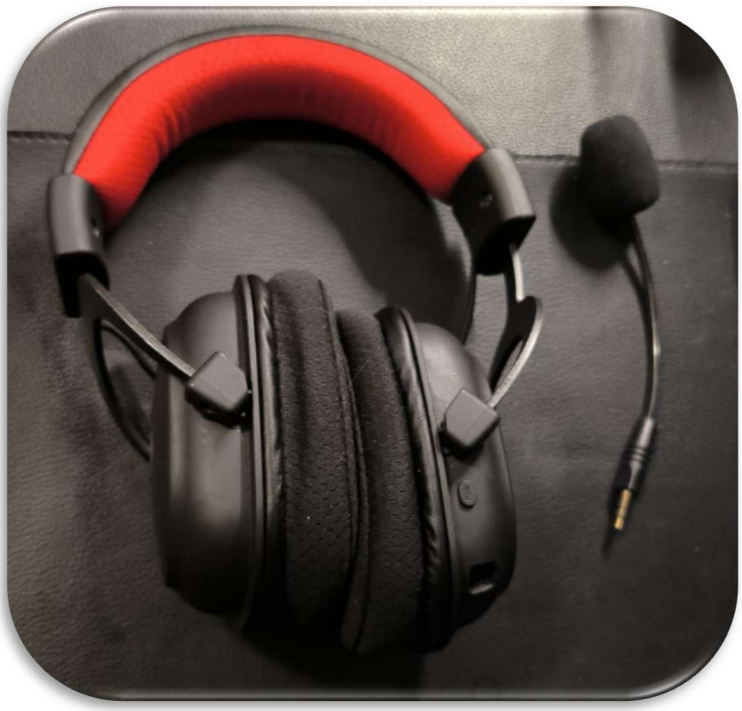
O desenvolvimento do sistema foi dividido em etapas bem definidas, permitindo a implementação, integração e validação gradual de cada módulo que compõe a arquitetura proposta.

3.1. Captura de áudio

A escolha do dispositivo de captação de áudio é um fator importante para o desempenho de sistemas de reconhecimento automático de fala, especialmente em ambientes com alto nível de ruído. Optou-se pela utilização de um *headset* com microfone integrado, em vez de microfones externos convencionais, visando maior estabilidade na captura do sinal de voz por ter mais proximidade física com a origem do sinal de voz, além de possibilitar implementações envolvendo saída de áudio como um aviso sonoro que sinaliza quando a palavra de ativação foi identificada.

Inicialmente, para a realização dos testes iniciais e validação do funcionamento do sistema de reconhecimento de voz, foi utilizado o headset Redragon Zeus Pro, apresentado na Figura 10, devido a sua disponibilidade no momento do desenvolvimento. Sua utilização permitiu verificar a integração do dispositivo de áudio com a plataforma embarcada além de avaliar o funcionamento dos algoritmos de captura e processamento de áudio sem a necessidade de aquisição imediata de equipamentos específicos.

Figura 10 - Headset de validação



Fonte: Do autor (2025)

Após a validação inicial, optou-se pela aquisição de um headset com melhores características técnicas, mais adequado ao contexto do projeto, buscando maior qualidade na captação da fala, melhor relação sinal-ruído e maior confiabilidade nos testes e na operação final do sistema.

O headset escolhido foi o Jabra Evolve 30 II, que possui microfone com Active Noise Cancellation (ANC), ou Cancelamento Ativo de Ruído, característica importante para reduzir a influência de sons indesejados do ambiente. Além disso, o dispositivo conta com placa de áudio integrada, requisito importante para o correto funcionamento do sistema, uma vez que permite a conversão do sinal analógico para digital diretamente no próprio *headset*, garantindo compatibilidade nativa com o sistema Linux embarcado e maior estabilidade na taxa de amostragem durante a captura do áudio. O modelo escolhido também apresenta resistência a respingos, fator relevante para que opere em uma embarcação.

Por fim, o modelo foi selecionado considerando também um bom equilíbrio entre desempenho e custo, atendendo às exigências técnicas do projeto sem ultrapassar o orçamento disponível. A Figura 11 apresenta o headset adquirido para o projeto proposto.

Figura 11 - Headset do projeto

Fonte: Do autor (2025)

3.2. Detecção da palavra de ativação

Para a detecção da palavra de ativação, foi utilizado um modelo de aprendizado de máquina treinado na plataforma Edge Impulse, voltado especificamente para execução em sistemas embarcados. Após o treinamento, o modelo foi exportado na forma de uma biblioteca C++ para Linux, gerada pelo Edge Impulse SDK, permitindo sua execução local na Raspberry Pi sem necessidade de conexão com a internet, com baixa latência e baixo consumo de recursos computacionais.

O módulo de detecção da palavra de ativação opera de forma contínua, analisando o fluxo de áudio capturado até identificar a ocorrência da palavra-chave definida. Somente após a detecção da *wake word* o sistema habilita o reconhecimento de comandos de voz, reduzindo ativações indevidas e processamento desnecessário.

3.3. Reconhecimento de fala

A escolha do Vosk neste projeto foi motivada por diversas vantagens específicas abaixo, como detalhado na documentação oficial do Vosk (ALPHA CEPHEI, 2025).

- Funcionamento 100% *offline*: essencial para o funcionamento em áreas remotas ou sem conectividade.
- Baixa latência: resposta relativamente rápida aos comandos, crucial em uma aplicação interativa.
- Resiliência a ruído: o sistema de reconhecimento do Vosk, mesmo utilizando modelos leves, apresenta desempenho satisfatório em ambientes com ruído ambiente, como motores e vento, comuns em embarcações.
- Baixo consumo computacional: o Vosk funciona de forma eficiente no Raspberry Pi 3 B+, sem exigir aceleração por GPU ou recursos dedicados.
- Flexibilidade e suporte a português: permite vocabulário personalizado e adaptações com relativa facilidade.

Embora Coqui STT tenha suporte a execução offline e modelos pré-treinados, sua estrutura de integração difere da abordagem baseada em bibliotecas C++ utilizada neste trabalho. Considerando que o Vosk oferece API nativa em C++ e integração direta à plataforma utilizada, optou-se por não utilizar o Coqui STT neste projeto.

Adicionalmente, Vosk possui modelos leves em português, como o utilizado neste projeto, com tempo de carregamento inferior a 3 segundos e consumo de memória abaixo de 100 MB para os modelos embarcados. O tempo de inferência médio foi inferior a 2 segundos para comandos curtos, aceitável para aplicações interativas.

3.4. Comunicação CAN

A arquitetura da embarcação solar em questão utiliza o CAN como protocolo principal de comunicação entre seus diversos módulos. O sistema desenvolvido neste trabalho atua como uma unidade de comando adicional, integrando-se a essa rede já existente. Para permitir a comunicação da Raspberry Pi com o barramento CAN, foi utilizado um *shield* com o transceptor MCP2515, já desenvolvido anteriormente pela equipe do barco solar. O MCP2515 é um controlador CAN com interface SPI utilizado em sistemas embarcados devido à sua compatibilidade com microcontroladores e microprocessadores (MICROCHIP, 2005).

A integração entre o controlador CAN MCP2515 e a Raspberry Pi foi realizada por meio da interface SPI, utilizando o subsistema *SocketCAN* do Linux para acesso à rede CAN e utilizando as bibliotecas padrão <*linux/can.h*> e <*linux/can/raw.h*> em linguagem C++. Essa abordagem permite acesso direto à pilha CAN do Linux, sem o uso de bibliotecas de alto nível, garantindo maior controle sobre a comunicação.

No contexto da rede CAN, o módulo MCV25 atua como nó transmissor, responsável pelo envio de mensagens de estado e de controle veicular. Não há, na lógica principal, processamento de mensagens recebidas, sendo a interpretação e execução dos comandos responsabilidade dos módulos da rede a quem se destina cada uma das mensagens.

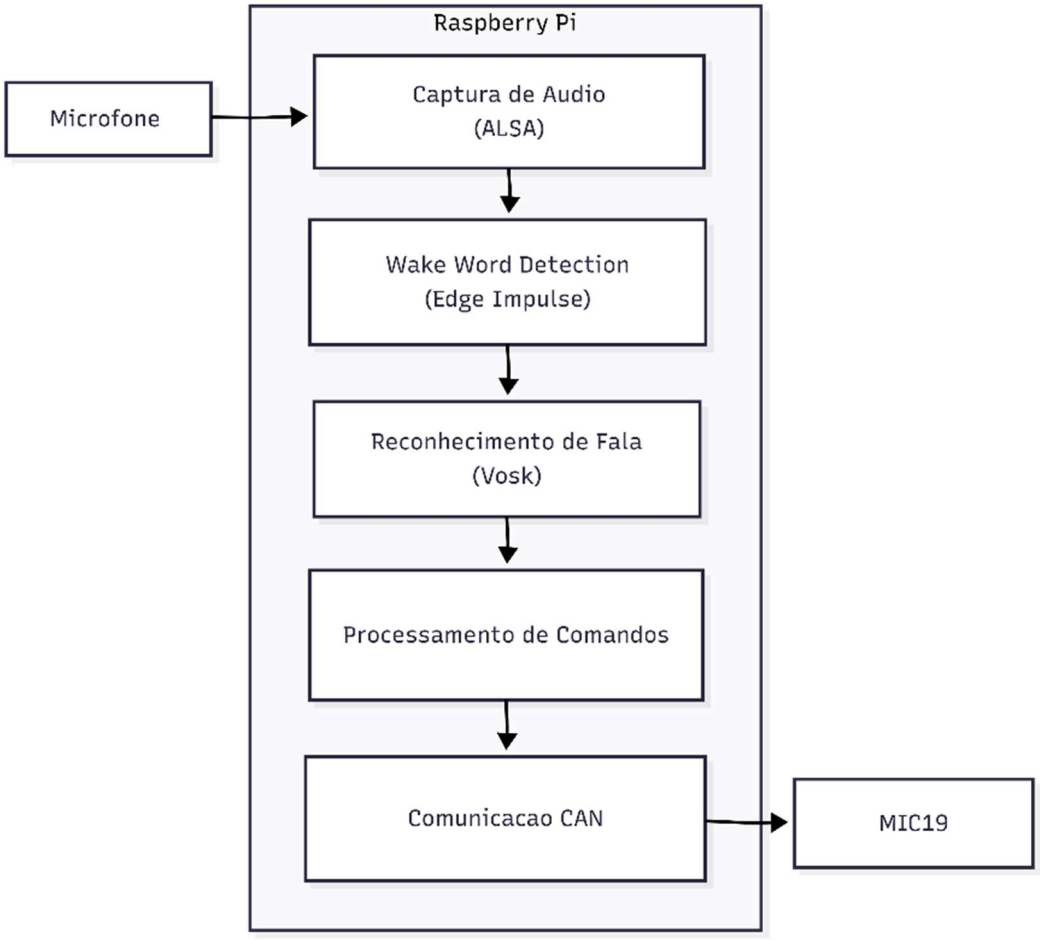
3.5. Arquitetura do sistema

A arquitetura geral do sistema de reconhecimento de voz, apresentado na Figura 12, evidencia o fluxo de dados desde a captura do áudio até o envio dos comandos para os demais módulos da embarcação. Nessa arquitetura, o sinal de áudio é inicialmente capturado pelo microfone do *headset* previamente apresentado e conectado ao sistema, sendo processado pelo subsistema ALSA, responsável pela interface de baixo nível com o hardware de áudio.

Em seguida, o áudio é encaminhado ao módulo de detecção da palavra de ativação (*wake word*), implementado por meio do Edge Impulse, que permanece em execução contínua aguardando a ativação do assistente. Após a detecção da *wake word*, o sistema passa a realizar o reconhecimento de fala propriamente dito utilizando a biblioteca Vosk, responsável pela transcrição dos comandos de voz.

O texto reconhecido é então interpretado pelo módulo de processamento de comandos, que converte as intenções do usuário em mensagens compatíveis com o protocolo da rede CAN. Essas mensagens são transmitidas ao MIC, responsável pela execução das ações correspondentes no sistema da embarcação, como controle de motor e direção. Esse encadeamento de etapas permite a integração do reconhecimento de voz ao sistema embarcado de forma estruturada e eficiente, mesmo sob restrições computacionais.

Figura 12 - Arquitetura Geral do Sistema



Fonte: Do autor (2025)

3.6. Shield CAN

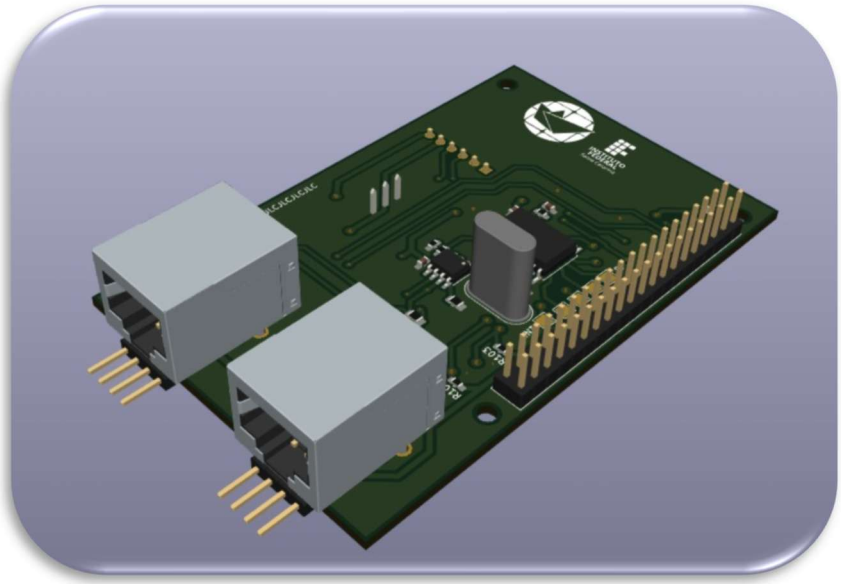
Pela limitação de hardwares disponíveis e como forma de simplificar o desenvolvimento do projeto, o assistente de voz opera no mesmo hardware da interface do painel e em paralelo, com serviços diferentes. Essa abordagem limita ainda mais os recursos computacionais disponíveis, visto que ambos dividirão estes recursos, porém nos permite utilizar somente um Shield CAN para ambas as aplicações.

A Figura 13 apresenta uma visualização 3D do Shield CAN desenvolvido pela equipe, o qual possibilita a comunicação entre a Raspberry Pi e a rede CAN da embarcação. Esse shield é compartilhado tanto pela interface do painel quanto pelo assistente de voz, uma vez que ambos utilizam a mesma Raspberry Pi como unidade de processamento, reduzindo a complexidade do hardware e a necessidade de

componentes adicionais, ao mesmo tempo em que garante a integração adequada entre os módulos de software e os demais sistemas da embarcação. Vale ressaltar que a barra de pinos presente na figura é para melhor visualização do componente, na placa utiliza-se uma barra de pinos fêmea 2x20 para compatibilidade com o *pinout* da Raspberry PI 3B+

Além disso, o Shield CAN realiza a conversão da energia fornecida para os níveis adequados exigidos pela Raspberry Pi, fornecendo uma tensão regulada de 5V e corrente suficiente para atender às demandas do sistema, garantindo operação estável mesmo em condições de variação de carga e ruído elétrico.

Figura 13 - Shield CAN



Fonte: Do autor (2025)

Ao longo das diferentes competições e iterações do projeto, Dado o ambiente hostil a sistemas eletrônicos em que opera pelo contexto de competições na água, o hardware sofreu alguns desgastes com tempo e algumas modificações foram realizadas no hardware com o objetivo de garantir o contínuo funcionamento do sistema, mantendo, contudo, a arquitetura funcional original. Este s*hield* é então conectado por meio do seu *pinout* lateral, que deve coincidir com o pinout da Raspberry Pi 3B+ utilizada, apresentada na Figura 14.

Figura 14 - Raspberry Pi model 3B+

Fonte: Do autor (2025)

Essa configuração permite que a Raspberry Pi seja alimentada utilizando a alimentação proveniente da embarcação, eliminando a necessidade de fontes externas adicionais. Além disso, o conjunto possibilita o acesso a recursos como a saída de vídeo, utilizada pela interface do painel, e à comunicação pela rede CAN, uma vez que o transceptor CAN encontra-se integrado e conectado à Raspberry Pi por meio da interface SPI.

Os conectores RJ45 presentes no *Shield CAN* desempenham um papel fundamental na arquitetura do sistema, atuando tanto como meio físico para a comunicação entre os módulos da embarcação por meio do protocolo CAN quanto como canal de distribuição de energia. Por meio desses cabos, os módulos recebem uma tensão de alimentação de 18V, ao mesmo tempo em que é viabilizada a transmissão de dados em altas frequências. Cabos do tipo CAT 5 e CAT 5e, utilizados na embarcação, suportam frequências de até 100 MHz, atendendo adequadamente aos requisitos de comunicação do sistema.

3.7. Processamento de Áudio

O processamento de áudio é uma etapa fundamental do sistema desenvolvido, pois impacta diretamente a confiabilidade da detecção da palavra de ativação e, consequentemente, de todo o fluxo de reconhecimento de comandos por voz. Nesta etapa, foram implementados mecanismos de aquisição, validação e preparação do sinal de áudio antes de verificar a palavra de ativação com o modelo gerado pela Edge Impulse.

3.7.1. Captura de áudio e validação inicial

A captura de áudio é realizada continuamente por meio da interface *Advanced Linux Sound Architecture* (ALSA), utilizando o microfone do *headset* conectado à Raspberry Pi. O áudio é inicialmente obtido com taxa de amostragem superior àquela exigida pelo modelo de *wake word* por conta do microfone utilizado, tornando necessária a aplicação de etapas intermediárias de validação e *downsample* do sinal.

Um dos primeiros cuidados foi a verificação de áudio nulo ou inválido. Durante os testes práticos, observou-se que situações como microfone desconectado, dispositivo em modo mudo ou falhas temporárias no subsistema de áudio resultavam na captura de *buffers* contendo valores constantes ou amplitudes extremamente baixas. Esses cenários, caso não tratados, poderiam levar o modelo a operar com dados inválidos, aumentando a incidência de erros no sistema.

Para mitigar esse problema, foi implementado um mecanismo de detecção de sinal constante. O algoritmo consiste em comparar todas as amostras de um *buffer* de áudio com a primeira amostra capturada. Caso todas as amostras apresentem exatamente o mesmo valor, ou seja, não exista variação ao longo do sinal, o trecho é classificado como inválido e descartado antes das etapas de pré-processamento e inferência. Essa abordagem possui baixo custo computacional, pois envolve apenas uma varredura linear no *buffer*, sendo adequada ao contexto do projeto.

3.7.2. Técnicas de *downsample* e filtros

Além da validação do sinal, foi necessário realizar o *downsampling* do áudio, uma vez que o modelo de *wake word* do Edge Impulse opera com taxa de amostragem

fixa de 16 kHz enquanto o hardware para gravação utiliza uma taxa de amostragem de 44.1 kHz.

Durante o desenvolvimento, foram avaliadas diferentes técnica de reamostragem. O primeiro método analisado foi o *nearest neighbor*, baseado na seleção direta de amostras do sinal original, caracterizado pelo baixo custo computacional. Em seguida, foi considerada a interpolação linear, que estima os valores do sinal em novos instantes a partir das amostras adjacentes, resultando em uma representação mais suave da forma de onda. Por fim, avaliou-se o método de média por blocos (*box averaging*), no qual cada amostra do sinal reamostrado é obtida a partir da média de um conjunto de amostras do sinal original, técnica equivalente à aplicação de um filtro passa-baixa simples seguido de decimação.

Esses três métodos de *downsampling* foram selecionados por apresentarem diferentes compromissos entre custo computacional e preservação das características espectrais do sinal de fala. A avaliação comparativa dessas abordagens é apresentada em capítulo posterior, considerando critérios de acurácia, latência e consumo computacional, de forma a embasar a escolha da estratégia mais adequada para o sistema embarcado proposto.

Também foram avaliadas técnicas de filtragem do sinal, como filtros passa-altas e passa-baixas, com o objetivo de reduzir ruídos de baixa frequência e interferências ambientais. No entanto, verificou-se que tais filtros apresentaram impacto limitado na qualidade perceptível do áudio e na acurácia da detecção da *wake word*, além de adicionarem carga computacional ao sistema embarcado. Considerando que o próprio modelo do Edge Impulse já aplica filtros internos e realiza extração de MFCC que são mais resistentes a ruído, decidiu-se não aplicar filtragens adicionais de forma permanente, mantendo o processamento de áudio o mais enxuto possível.

Todas as etapas de tratamento do áudio foram projetadas para operar de forma a garantir uma aplicação interativa, garantindo que o fluxo contínuo de dados não prejudicasse a experiência do usuário referente ao tempo de resposta do sistema. O áudio validado e reamostrado é então encaminhado ao modelo de detecção da palavra de ativação, garantindo maior resiliência do sistema para falhas de hardware, ruídos ambientais e limitações da plataforma embarcada.

3.8. Treinamento do modelo para *Wake Word Detection*

A detecção da palavra de ativação constitui a segunda etapa da arquitetura geral, sendo responsável por identificar, de forma contínua e eficiente, a ocorrência da palavra-chave definida para ativação do sistema. Essa abordagem permite reduzir ativações indevidas e otimizar o uso dos recursos computacionais da plataforma embarcada, uma vez que o reconhecimento completo de comandos de voz só é iniciado após a confirmação da palavra.

3.8.1. Base de dados

A base de dados utilizada para o treinamento da rede neural foi composta por áudios reais e sintéticos, visando aumentar a acurácia do modelo frente às variações comuns em ambientes embarcados. Foram utilizados áudios gravados por diferentes pessoas pronunciando a palavra de ativação, bem como amostras de ruído e fala irrelevante (*noise* e *unknown*), representativas do ambiente de operação da embarcação.

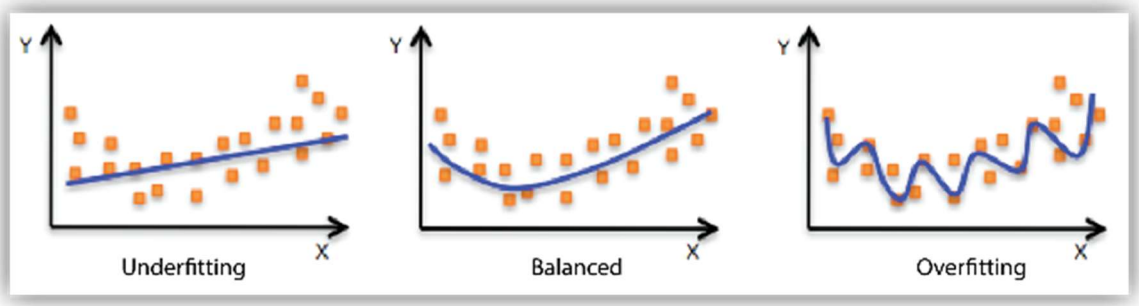
Além dos áudios gravados manualmente, foram utilizados áudios gerados por inteligência artificial, os quais permitiram ampliar o conjunto de dados sem a necessidade de novas gravações físicas. Para tornar o treinamento mais realista, foram aplicadas técnicas de aumento de dados (*data augmentation*), incluindo a adição de ruídos, variações de ganho, distorções leves e alterações na qualidade do sinal, simulando condições típicas de captura por microfones, como interferências ambientais, variações de distância e diferenças de timbre entre as vozes.

Essa combinação de dados reais e sintéticos contribuiu para reduzir o risco de sobreajuste (*overfitting*) e melhorar a capacidade de generalização do modelo. No contexto de aprendizado de máquina, *overfitting*, representado na Figura 15, ocorre quando um modelo se ajusta de forma excessiva aos exemplos de treinamento, aprendendo detalhes ou ruídos específicos desse conjunto e, por consequência, não consegue desempenhar bem em dados não vistos durante o treinamento. Em outras palavras, um modelo *overfitted* apresenta bom desempenho nos dados de treinamento, mas falha ao generalizar para novos dados, justamente por “memorizar” o conjunto original em vez de aprender padrões representativos do problema como um todo (AWS, 2025)

Por outro lado, o *underfitting* ocorre quando o modelo apresenta desempenho insatisfatório já nos próprios dados de treinamento. De acordo com a AWS (2025), o *underfitting* acontece quando o modelo apresenta baixo desempenho nos próprios dados de treinamento, evidenciando que não conseguiu capturar adequadamente a relação entre as entradas e as saídas esperadas. Esse comportamento geralmente está associado a modelos excessivamente simples ou com baixa capacidade de representação, que não conseguem descrever a complexidade do problema. Como consequência, o desempenho tende a ser igualmente baixo tanto em dados de treinamento quanto em dados de validação ou teste.

Entre esses dois extremos encontra-se o modelo balanceado (*Balanced*), no qual há bom desempenho nos dados de treinamento e desempenho semelhante em dados de validação ou teste. Esse equilíbrio indica que o modelo conseguiu aprender padrões relevantes do problema, mantendo capacidade de generalização para novos exemplos. No contexto deste trabalho, o uso combinado de dados reais, dados sintéticos e técnicas de aumento de dados contribuiu para aproximar o modelo desse ponto de equilíbrio, reduzindo tanto o risco de *overfitting* quanto a probabilidade de *underfitting*, ao ampliar a diversidade e representatividade do conjunto de treinamento.

Figura 15 - *Underfitting* vs. *Overfitting*



Fonte: AWS (2025)

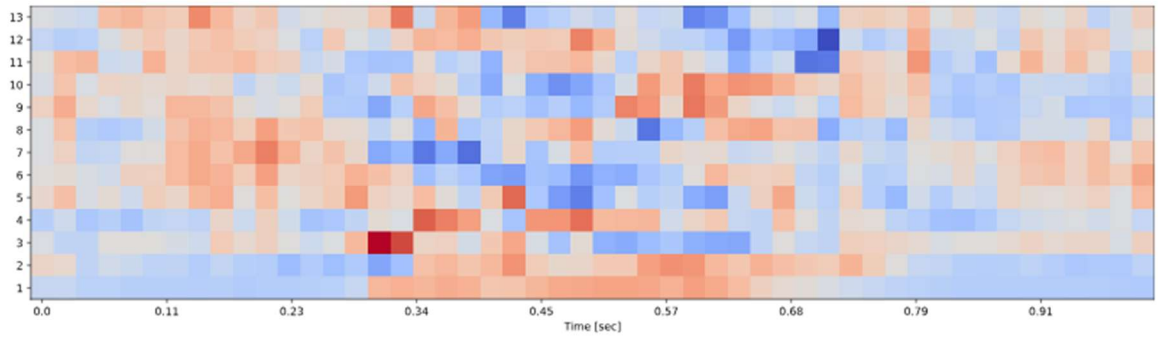
3.8.2. Extração de características

Antes do treinamento da rede neural, os sinais de áudio foram processados para a extração de características relevantes. Conforme mencionado anteriormente, o Edge Impulse utiliza os coeficientes cepstrais em escala Mel (MFCC), usados em

sistemas de reconhecimento de fala por representarem de forma eficiente as características espectrais do áudio perceptíveis ao ouvido humano.

A Figura 16 apresenta a visualização dos coeficientes MFCC extraídos de uma amostra de áudio utilizada no treinamento da palavra de ativação. No gráfico, o eixo horizontal representa o tempo do sinal de áudio, segmentado em janelas sucessivas, enquanto o eixo vertical corresponde ao índice dos coeficientes MFCC extraídos em cada janela. A variação de cores indica a magnitude dos coeficientes, indicando como a distribuição espectral do sinal se altera ao longo do tempo. Coeficientes azuis indicam valores negativos e vermelhos indicam valores positivos, onde tons mais fortes representam valores absolutos mais distantes do zero.

Figura 16 - *Cepstral Coefficients*



Fonte: Do autor (2025)

Essa representação tempo–frequência permite destacar padrões espectrais e temporais característicos da palavra de ativação, facilitando a identificação de assinaturas acústicas específicas e auxiliando a rede neural no processo de classificação.

3.8.3. Treinamento da Rede Neural

O modelo de *wake word detection* foi treinado pela Edge Impulse utilizando uma rede neural leve, adequada à execução em dispositivos embarcados. Durante o treinamento, o modelo aprendeu a distinguir entre três classes principais: a palavra de ativação (Zenira), ruídos ambientes (*noise*) e falas irrelevantes (*unknown*).

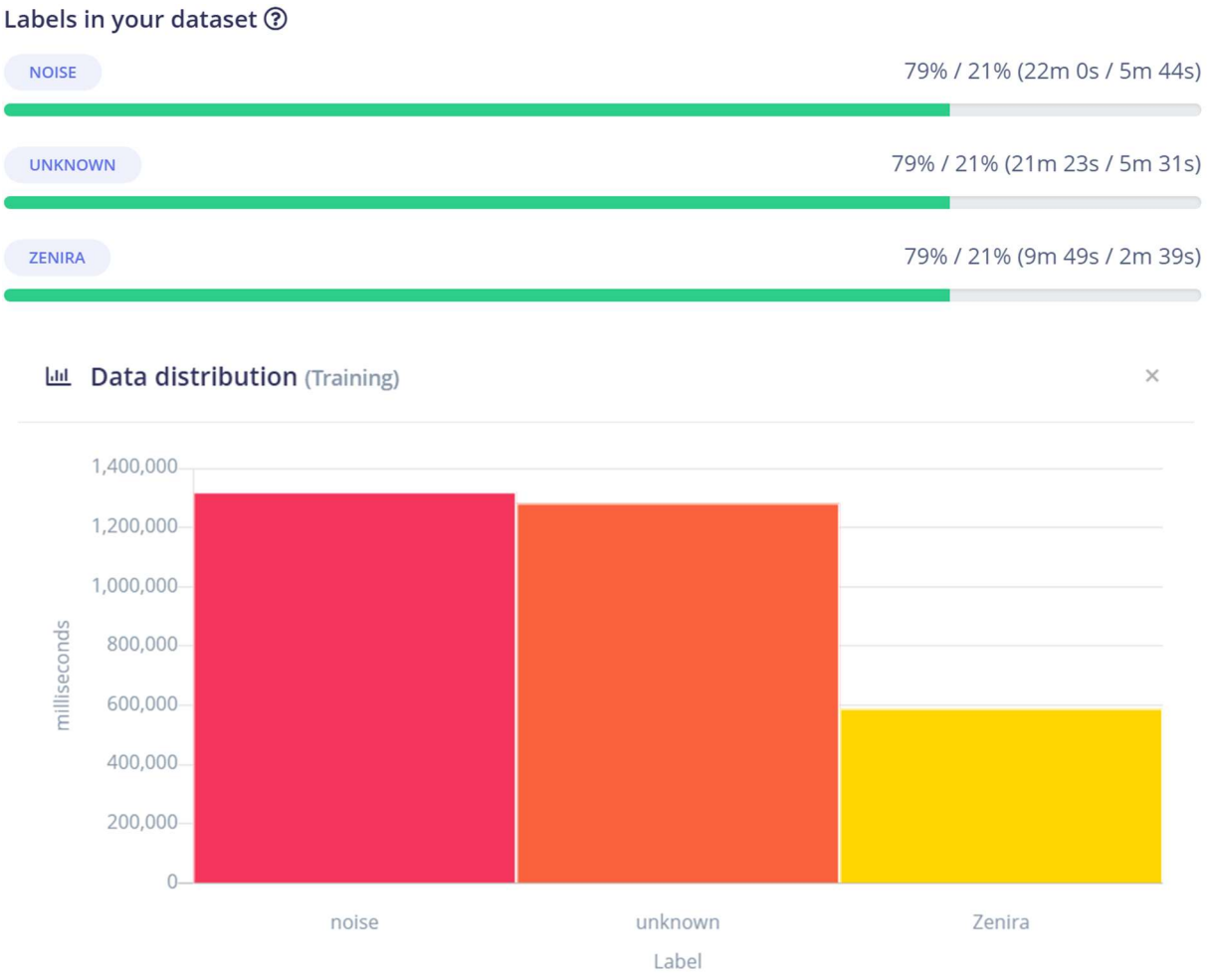
O processo de treinamento foi realizado de forma supervisionada, a partir de um conjunto de dados previamente rotulado. O *dataset*, apresentado na Figura 17, foi

composto por aproximadamente 1 hora e 7 minutos de áudios combinando as 3 categorias, separados entre 79% para treinamento e 21% para testes e validação, conforme recomendado pela plataforma.

Idealmente, o conjunto de dados da palavra-chave “Zenira” deveria possuir uma quantidade de amostras equivalente às categorias *unknown* e *noise*, garantindo um treinamento mais balanceado. A coleta de áudios com a palavra de ativação é um processo lento que exige gravações controladas e repetidas da palavra. Ainda assim, a quantidade obtida foi suficiente para um desempenho satisfatório do modelo, não sendo necessária uma ampliação excessiva do conjunto.

Os áudios rotulados como *unknown* e *noise* foram obtidos a partir de *datasets* gratuitos, desenvolvidos especificamente para testes de sistemas de detecção de palavras de ativação, contribuindo para a diversidade do treinamento.

Figura 17 - Composição do *Dataset*



Fonte: Do autor (2025)

Durante o processo, o desempenho do modelo foi monitorado para que fosse possível realizar ajustes nos parâmetros do modelo até alcançar uma taxa de acerto razoável. A avaliação do desempenho do modelo foi realizada por meio da matriz de confusão, apresentada na Figura 18. Essa representação permite analisar de forma detalhada os acertos e erros de classificação para cada classe, evidenciando possíveis confusões entre a palavra de ativação e outros sons.

Figura 18 - Matriz de confusão

	ZENIRA	NOISE	UNKNOWN	UNCERTAIN
ZENIRA	**89.3%**	1.9%	6.9%	1.9%
NOISE	0.3%	**93.0%**	5.5%	1.2%
UNKNOWN	3.6%	2.7%	**93.4%**	0.3%
F1 SCORE	0.90	0.95	0.92	

Fonte: Do autor (2025)

A matriz de confusão indica que o modelo foi capaz de identificar corretamente a maior parte das ocorrências da palavra de ativação, mantendo uma baixa taxa de falsos positivos, o que evita ativações indesejadas do sistema em ambiente real.

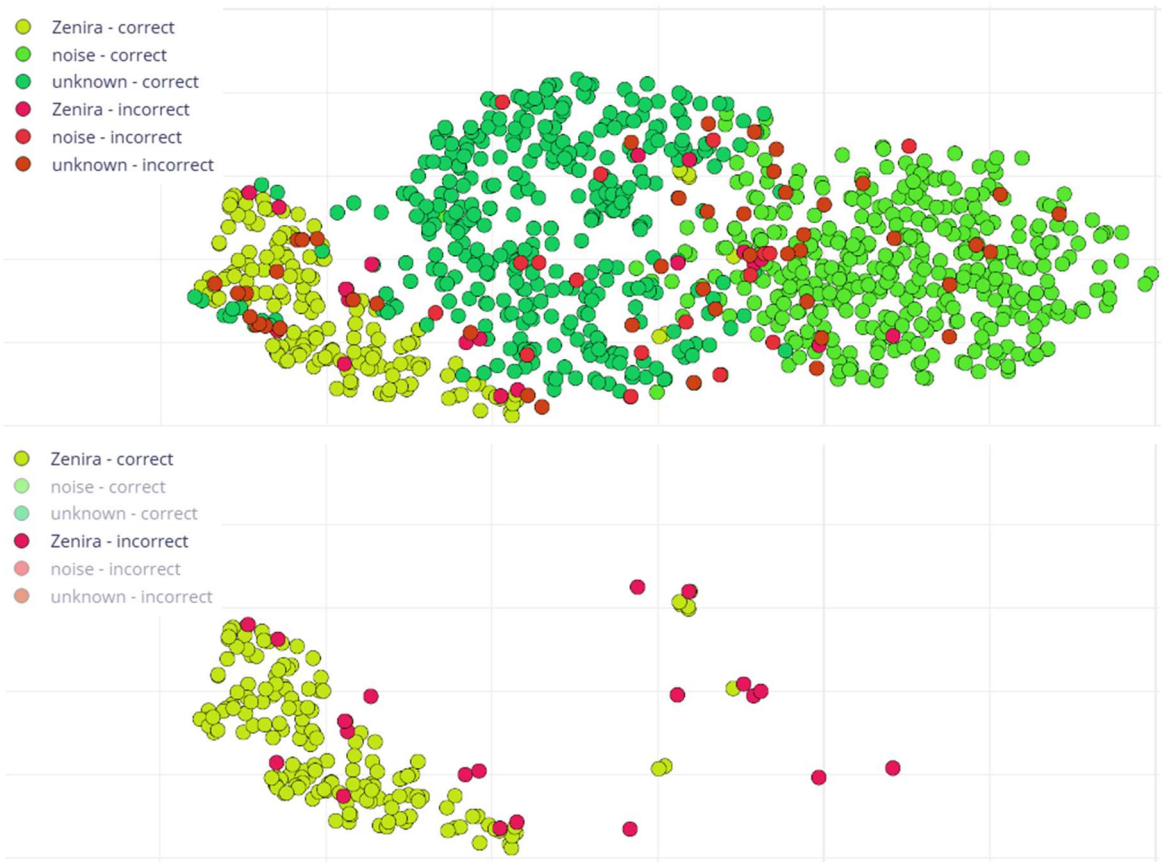
A Figura 19 apresenta a visualização da distribuição das amostras de dados classificadas pela rede neural após o treinamento seguida da mesma visualização filtrando apenas amostras referentes a palavra de ativação. Cada ponto representa uma amostra individual após a etapa de extração de características, sendo sua posição determinada por uma técnica de redução de dimensionalidade que projeta os dados de alta dimensão em duas componentes principais. Essa visualização ajuda a analisar a separação entre as classes aprendida pelo modelo, assim como a presença de algumas regiões de sobreposição que podem indicar similaridade acústica entre determinadas classes.

É possível observar a formação de um agrupamento bem definido correspondente à palavra de ativação, caracterizado pela alta densidade e baixa dispersão interna, indicando que o modelo conseguiu capturar padrões acústicos consistentes associados à palavra. A separação espacial desse agrupamento em

relação às demais classes indica boa separabilidade entre a palavra de ativação e os demais sons presentes no conjunto de dados.

Embora a figura indique separação satisfatória entre as classes principais, ainda é possível observar regiões com sobreposição, sugerindo que o espaço de características aprendido pelo modelo pode ser melhorado com a ampliação do conjunto de dados, aplicação de técnica de *data augmentation* e ajustes na arquitetura ou nos parâmetros do modelo. Nota-se também a presença de pequenos agrupamentos da palavra de ativação inseridos em regiões onde temos predominantemente *noise* e *unknown*.

Figura 19 - Classificação das amostras pela rede neural



Fonte: Do autor (2025)

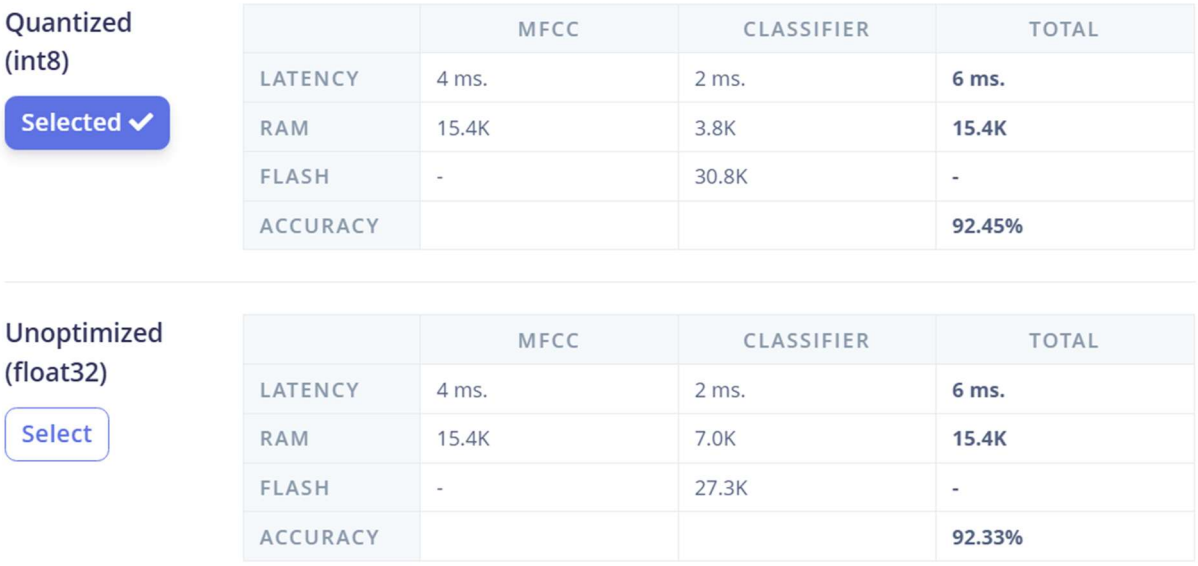
Após o treinamento, o modelo foi utilizado no sistema embarcado na forma de uma biblioteca C++ otimizada, gerada por meio do EON Compiler da plataforma Edge Impulse. Essa abordagem permitiu a execução contínua do modelo na Raspberry Pi, sem comprometer significativamente o desempenho do sistema, atendendo aos requisitos de tempo real e baixo consumo de recursos computacionais.

A etapa de detecção da palavra de ativação mostrou-se fundamental para o funcionamento eficiente do assistente por voz, atuando como um filtro inicial que garante que apenas comandos intencionais do piloto sejam processados nas etapas subsequentes do sistema.

3.8.4. Quantização do modelo

A Figura 20 apresenta a comparação entre o modelo não quantizado, representado em ponto flutuante de 32 bits (float32), e o modelo quantizado com representação inteira de 8 bits (int8), conforme disponibilizado na seção de *deploy* da plataforma Edge Impulse. A análise considera métricas relevantes para sistemas embarcados, como latência, uso de memória RAM, ocupação de memória Flash e acurácia do modelo.

Figura 20 - Quantização do modelo



Fonte: Do autor (2025)

Observa-se que ambos os modelos apresentam a mesma latência total de inferência, estimada em aproximadamente 6 ms, sendo composta pelo tempo de extração de características MFCC e pelo tempo de execução do classificador. Esse resultado indica que, no hardware avaliado, a quantização não impactou significativamente o tempo de processamento.

Em relação ao uso de memória, verifica-se uma redução no consumo de RAM pelo classificador no modelo quantizado, passando de aproximadamente 7,0 kB no modelo float32 para cerca de 3,8 kB no modelo int8. Essa diminuição é consistente com o esperado, já que a quantização reduz a precisão numérica dos pesos e ativações, resultando em estruturas de dados mais compactas e adequadas para dispositivos com recursos limitados, mas a diferença é praticamente irrelevante frente ao hardware adotado para o projeto.

Um ponto interessante observado na figura é o leve aumento da acurácia do modelo quantizado, que passou de 92,33% no modelo não quantizado para 92,45% no modelo quantizado. É importante ressaltar que estes valores incluem os erros de leitura entre as categorias *noise* e *unknown* que para o nosso caso são irrelevantes. Os valores relevantes para este projeto podem ser encontrados na matriz de confusão (Figura 18) que passaram de 88,7% no modelo não quantizado para 89,3% no modelo quantizado.

A pequena variação de acurácia observada entre o modelo quantizado e o modelo não quantizado pode ser atribuída a diversos fatores. Esse comportamento pode estar relacionado, entre outros aspectos, à forma como a plataforma Edge Impulse realiza o processo de quantização, que ocorre na etapa de *deploy* do modelo, sem necessariamente incluir o treinamento completo com pesos quantizados, o que pode introduzir diferenças sutis no comportamento.

Além disso, o tamanho e a representatividade do conjunto de dados utilizado influenciam diretamente na estabilidade das métricas de desempenho. Conforme discutido por Althnian *et al.* (2021), não existe um tamanho ótimo universal de *dataset*, sendo a performance do modelo fortemente dependente de quão bem os dados de treinamento representam a distribuição real do sinal de entrada. Em conjuntos de dados reduzidos ou com variabilidade limitada, pequenas oscilações de acurácia entre diferentes versões do modelo são esperadas e não indicam, necessariamente, erro de treinamento ou falha de convergência.

Dessa forma, os resultados dos testes indicam que a quantização do modelo não apenas atende aos requisitos de redução de consumo de memória para sistemas embarcados, como também mantém e neste caso melhora o desempenho em termos de acurácia, se mostrando adequada para aplicações de reconhecimento de voz em dispositivos com restrições computacionais.

3.9. Implementação do Reconhecimento por Voz

A instalação do Vosk foi realizada diretamente na plataforma Raspberry Pi, utilizando a versão pré-compilada v0.3.45 compatível com a arquitetura ARMv7. Embora a compilação manual da biblioteca Kaldi possa permitir otimizações específicas para a aplicação, esse processo envolve alta complexidade técnica, dependências extensivas e elevado tempo de compilação.

No contexto deste projeto, cujo foco estava na implementação e validação do sistema embarcado de reconhecimento de fala, optou-se pela utilização da versão pré-compilada, já otimizada para arquiteturas ARM, garantindo estabilidade, compatibilidade e redução do tempo de desenvolvimento. Essa abordagem permitiu concentrar esforços na integração com o sistema de captura de áudio, processamento de comandos e comunicação via rede CAN.

Além da biblioteca principal de reconhecimento de fala Vosk, foram configuradas as dependências necessárias para a captura e o processamento de áudio em ambiente Linux embarcado, com destaque para o subsistema ALSA e a biblioteca *libasound*, responsáveis pelo acesso ao dispositivo de áudio, configuração dos parâmetros de amostragem e leitura contínua do sinal acústico. Também foram utilizadas bibliotecas auxiliares como *vector* para armazenamento e manipulação dos dados e *cmath* para operações matemáticas dos processos de processamento digital de sinais.

O fluxo de áudio capturado via ALSA foi segmentado em blocos e encaminhado diretamente ao objeto *Recognizer*, responsável pela inferência em tempo real. A inicialização do modelo incluiu o carregamento explícito dos arquivos acústicos e linguísticos em memória, seguido da criação das estruturas de reconhecimento.

O gerenciamento manual dos *buffers* de áudio possibilitou controle preciso sobre tamanho de amostras, taxa de amostragem (16 kHz) e tratamento de falhas na captura. Os resultados retornados pela API, em formato JSON, foram processados para extração do texto reconhecido e posterior envio dos comandos via rede CAN. Para garantir estabilidade durante a execução contínua do sistema, foram implementados mecanismos de inicialização segura e verificação do correto carregamento do modelo acústico e linguístico, prevenindo falhas por conta de arquivos ausentes ou incompatíveis.

3.9.1. Criação do modelo de comandos personalizados

Buscando aumentar a confiabilidade do reconhecimento e reduzir o número de inferências incorretas, foi utilizado um modelo de comandos personalizados, configurado especificamente para o conjunto definido de comandos necessários para o projeto. Diferente de um modelo de linguagem genérico, essa abordagem limita o vocabulário reconhecido pelo sistema, tornando-o mais resistente a ruídos ambientais e variações de pronúncia. A definição desse vocabulário restrito também auxilia na redução do tempo de processamento e no aumento da taxa de acerto, já que o mecanismo de decodificação passa a operar sobre um espaço de busca significativamente menor.

O modelo foi configurado utilizando as funcionalidades do Vosk para reconhecimento baseado em gramática, onde um conjunto previamente definido de palavras e expressões, representado na Tabela 2, é fornecido ao mecanismo de reconhecimento. Essa lista inclui comandos relacionados à navegação e controle da embarcação, como mudanças de direção, ajustes de velocidade e comandos auxiliares.

Tabela 2 - Comandos aceitos

Categoria	***Comando de voz***	***Valor***	***Ação***
Velocidade	*Mudar velocidade para <valor> Velocidade para <valor> Velocidade <valor>*	*0 – 100% (Incrementos de 10%)*	*Ajusta duty cycle do motor*
Velocidade	*[Ligar / Desligar] motor Motor [On / Off]*	*-*	*Liga / Desliga motor*
Estado Geral	*[Ligar / Desligar] barco Barco [On / Off]*	*-*	*Liga / Desliga barco*
Direção	*Virar a [Esquerda / Direita] [Esquerda / Direita]*	*-*	*Incrementa 10° ao atual valor de direção*
Direção	*Virar <valor> graus a <direção> <valor> graus a <direção>*	*10° - 90° (Incrementos de 10°)*	*Incrementa <valor>° ao atual valor de direção*
Direção	*Seguir reto Reto Centralizar*	*-*	*Centraliza a direção (0°)*

Fonte: Do autor (2025)

No controle de velocidade do motor, são aceitos valores entre 0% e 100%, com incrementos de 10%, conforme implementado no mapeamento interno do sistema. Esses valores correspondem diretamente ao duty cycle aplicado ao motor. Para o controle direcional da rabeta, são aceitos ângulos entre 0° e 90° para cada lado, também com incrementos de 10°. A direção é codificada como 0 (bombordo/esquerda) ou 1 (estibordo/direita).

Como forma de padronizar o envio de mensagens pela rede CAN, um arquivo *can_ids.h*, comum a todos os módulos da embarcação que utilizam a rede CAN, declara o endereço e tamanho das mensagens que poderão ser enviadas na rede, além de outras informações úteis para o envio e recebimento destas mensagens como o conteúdo de cada byte da mensagem e a sua frequência de envio.

As mensagens do MCV25 são enviadas com a mesma assinatura, indicando que foram enviadas por ele, porém cada categoria de mensagem está vinculada a um ID de mensagem diferente, conforme indicado na Tabela 3.

Tabela 3 - Identificador das mensagens

Categoria	***Assinatura***	***Identificador***	***Tamanho (bytes)***
Velocidade	242	91	4
Direção	242	92	3
Estado Geral	242	93	2

Fonte: Do autor (2025)

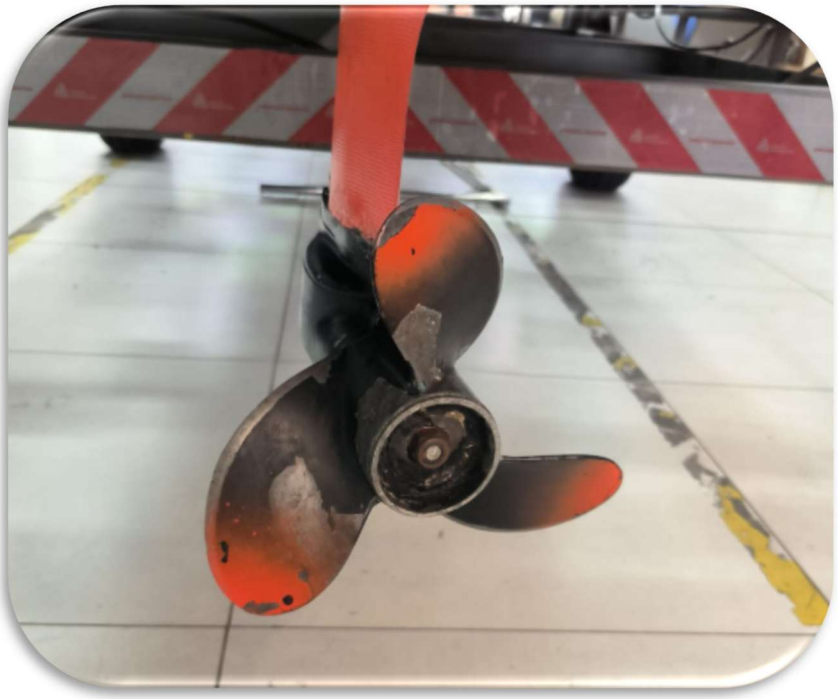
3.9.2. Controle da embarcação

Uma vez codificados e transmitidos pela rede CAN, os comandos de voz são interpretados pelo módulo responsável pelos atuadores da embarcação, resultando em ajustes reais no barco. Assim, parâmetros como percentual de velocidade e ângulo de navegação passam a controlar diretamente o motor e a rabeta, convertendo instruções digitais em ações físicas que determinam o deslocamento da embarcação.

A Figura 21 apresenta a rabeta da embarcação, componente responsável pela conversão dos comandos de direção em variações no ângulo de navegação,

permitindo o controle do sentido de deslocamento. O ajuste é realizado de forma incremental em relação ao ângulo atual da rabeta. Assim, caso a rabeta esteja posicionada, por exemplo, totalmente à direita, um comando para virar 90° à esquerda resulta no posicionamento central da rabeta. Adicionalmente, o sistema aceita o comando de centralização, no qual o ângulo da rabeta é ajustado diretamente para 0°, independentemente de sua posição atual.

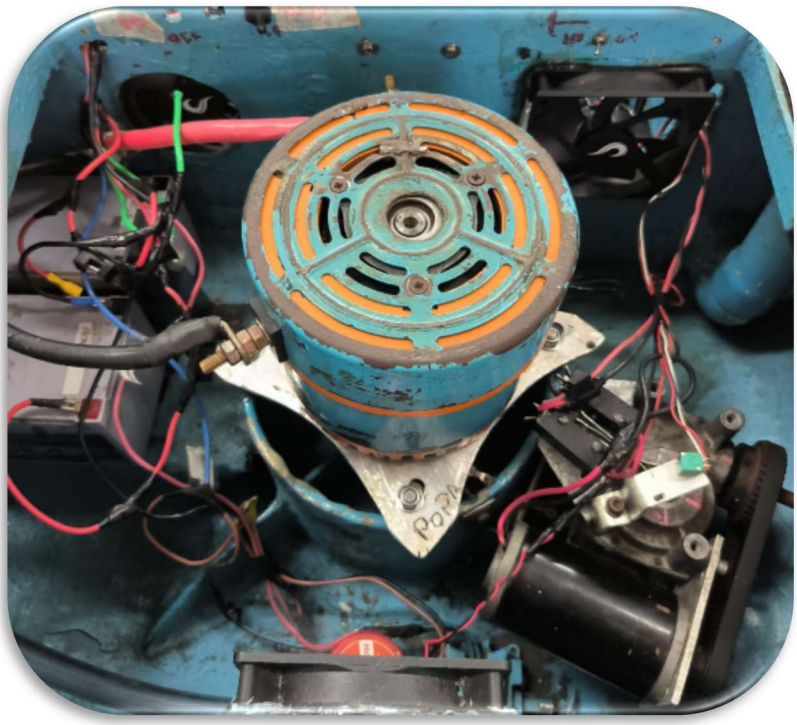
Figura 21 - Rabeta e hélice



Fonte: Do autor (2025)

Já a Figura 22 apresenta o motor da embarcação, responsável pela geração do torque necessário para o deslocamento. Diferentemente do controle de direção, que opera de forma incremental em relação à posição atual da rabeta, o ajuste de velocidade do motor é realizado de maneira absoluta. Isso significa que o valor solicitado por meio do comando de voz substitui diretamente o *duty cycle* atualmente aplicado, independentemente do estado anterior.

Figura 22 - Motor e baterias auxiliares



Fonte: Do autor (2025)

3.9.3. Integração do Vosk

O Vosk foi integrado ao sistema de forma complementar ao mecanismo de detecção da palavra de ativação. Após a identificação da palavra, o sistema passa para um modo ativo de escuta, no qual o fluxo de áudio é direcionado ao reconhecedor de fala do Vosk. Esse reconhecimento ocorre de forma contínua durante a janela de tempo definida de cinco segundos, permitindo que o comando completo seja capturado e interpretado.

O Vosk retorna uma string com a fala transcrita e esse texto é então analisado por um módulo de interpretação de comandos, responsável por mapear a fala reconhecida para ações específicas, como o envio de mensagens pela rede CAN para os demais módulos da embarcação.

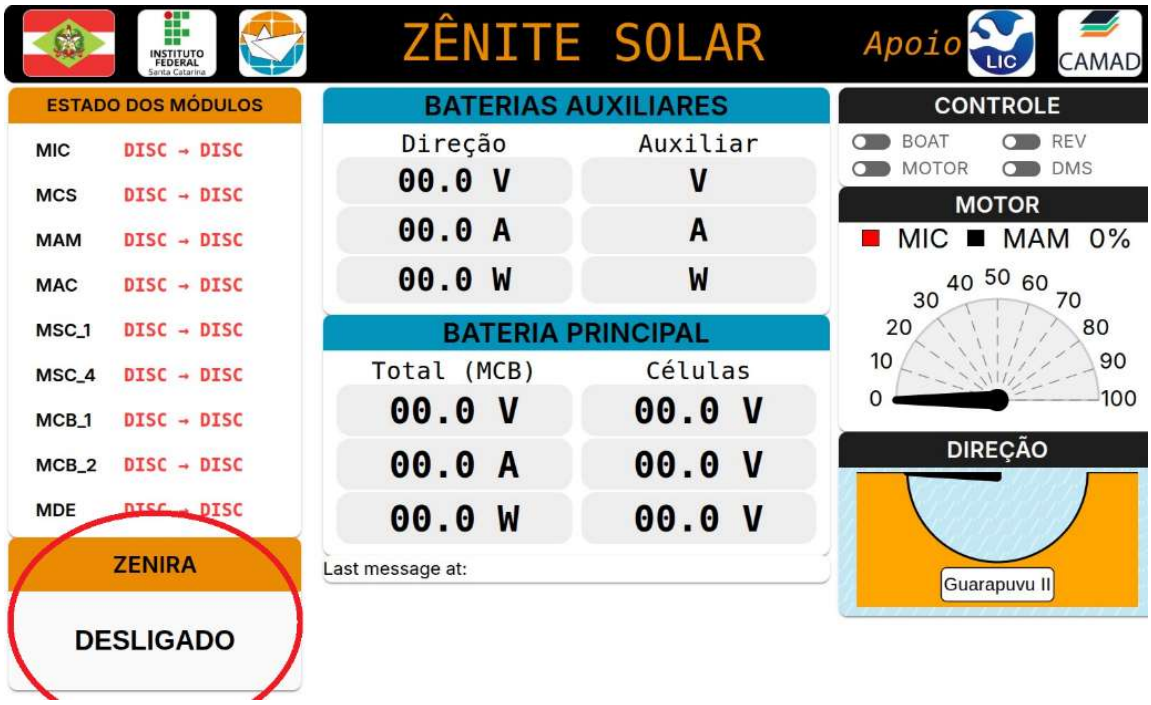
Foram implementados também mecanismos de controle de estado e tratamento de erros, garantindo que falhas pontuais no reconhecimento ou na captura de áudio não comprometam o funcionamento geral do sistema. Caso o reconhecimento não resulte em um comando válido, o sistema retorna automaticamente ao estado de escuta da palavra de ativação.

3.10. Painel e Interface

A interface do painel da embarcação, apresentada na Figura 23, foi mantida praticamente inalterada em relação à versão original do projeto. O painel já desempenhava sua função principal, que é apresentar ao piloto informações relevantes dos diferentes módulos da embarcação, permitindo o monitoramento em tempo real de parâmetros como a tensão das baterias, a direção da rabeta, a velocidade do motor e o estado de outros subsistemas.

Para o projeto, foi adicionado à interface um novo card dedicado ao assistente por voz, destacado pelo círculo vermelho, para informar de maneira clara o estado atual do sistema de reconhecimento de voz. Os estados possíveis são “desligado”, “aguardando” e “escutando”. O estado escutando é ativado somente após o reconhecimento da palavra de ativação e, nesse momento, o card altera sua cor de fundo para azul e um sinal sonoro é emitido no headset, facilitando a identificação visual e sonora de que o sistema está pronto para receber comandos de voz.

Figura 23 - Interface do painel



Fonte: Do autor (2025)

Além disso, foi realizada a integração do assistente por voz com um controle físico já existente no painel apresentado na Figura 24. A chave do painel, indicada pelo círculo vermelho, não estava associada a nenhuma funcionalidade específica, portanto foi reaproveitada para habilitar ou desabilitar o funcionamento da assistente por voz Zenira, alterando o estado do sistema de “desligado” para “aguardando”. Para isso, o código do MIC foi modificado de modo que os comandos de voz só sejam interpretados quando essa chave estiver ativa.

Figura 24 - Interface do painel

Fonte: Do autor (2025)

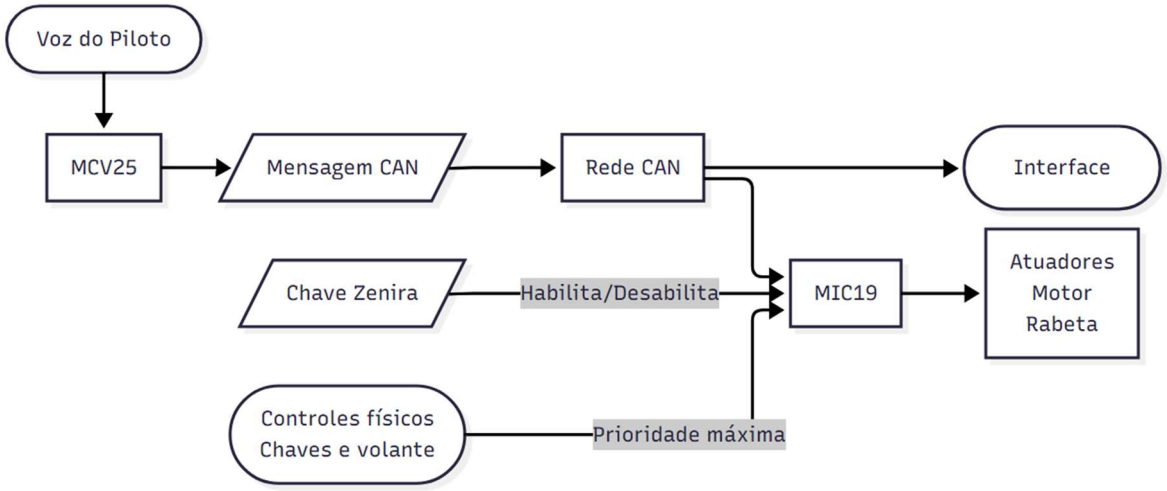
Essa abordagem aumenta a segurança operacional do sistema, evitando ativações indesejadas do assistente por voz, além de oferecer ao piloto maior controle sobre quando a interface por voz deve estar disponível durante a operação da embarcação.

3.11. Integração dos módulos

A integração entre os módulos do sistema, representado na Figura 25, foi realizada por meio da rede CAN da embarcação. Nesta integração, o MCV é responsável exclusivamente pela interpretação dos comandos de voz reconhecidos pelo sistema. Após o processamento, ele gera como única saída mensagens CAN

específicas, que representam as intenções de comando do piloto, como alteração de velocidade ou direção.

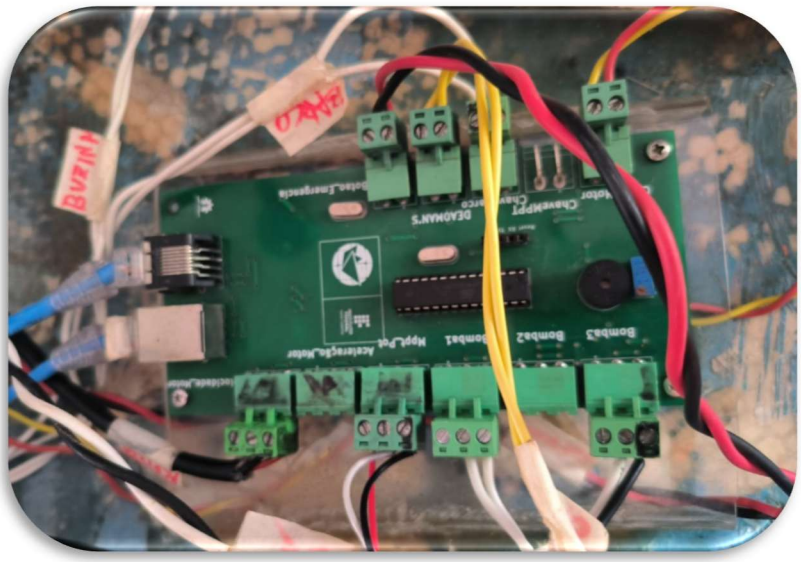
Figura 25 - Integração dos módulos



Fonte: Do autor (2025)

Para possibilitar a integração com o sistema de reconhecimento de voz, foi necessária a modificação do firmware do MIC19, apresentado na Figura 26, a fim de permitir o reconhecimento das mensagens provenientes do MCV25. Essas alterações envolveram a inclusão do novo módulo nas filtragens de mensagens CAN aceitas pelo sistema, bem como a implementação de funções específicas para o tratamento de cada comando recebido. Durante esse processo, foi considerada a prioridade das ações físicas do piloto sobre os comandos enviados via voz, garantindo a segurança e a confiabilidade da operação da embarcação.

Figura 26 - MIC19



Fonte: Do autor (2025)

O módulo MIC atua como intermediário entre as mensagens do MCV25 e os atuadores da embarcação. Ao receber comandos via rede CAN, o MIC as interpreta e traduz em ações concretas para os demais subsistemas, como acionamento do barco, controle de aceleração e ajuste da direção da rabeta. Essa separação de responsabilidades contribui para a modularidade do sistema e facilita futuras manutenções.

O processamento dos comandos do assistente por voz está condicionado ao estado da chave física que habilita a Zenira. Quando essa chave está desativada, o MIC ignora completamente as mensagens CAN do MCV, impedindo que comandos de voz sejam executados de forma não intencional.

Mesmo quando habilitado, o sistema foi projetado para garantir prioridade total às ações físicas do piloto. Os comandos realizados por meio das chaves físicas e do volante podem, a qualquer momento, sobrescrever os valores definidos por comandos de voz. Por exemplo, caso a velocidade seja ajustada para um determinado valor via assistente de voz e, em seguida, modificada manualmente pelo piloto, o valor definido pela interface física prevalece. Após essa intervenção manual, o piloto pode novamente utilizar comandos de voz para alterar os parâmetros, se assim desejar. Essa lógica garante que o controle direto do piloto tenha sempre prioridade, aumentando a segurança e a confiabilidade da embarcação.

4. TESTES E RESULTADOS

Este capítulo apresenta os testes experimentais adotados para a avaliação do sistema desenvolvido, além dos principais resultados obtidos a partir deles. As análises realizadas visam verificar o comportamento do sistema em diferentes condições de operação, considerando aspectos funcionais e de desempenho.

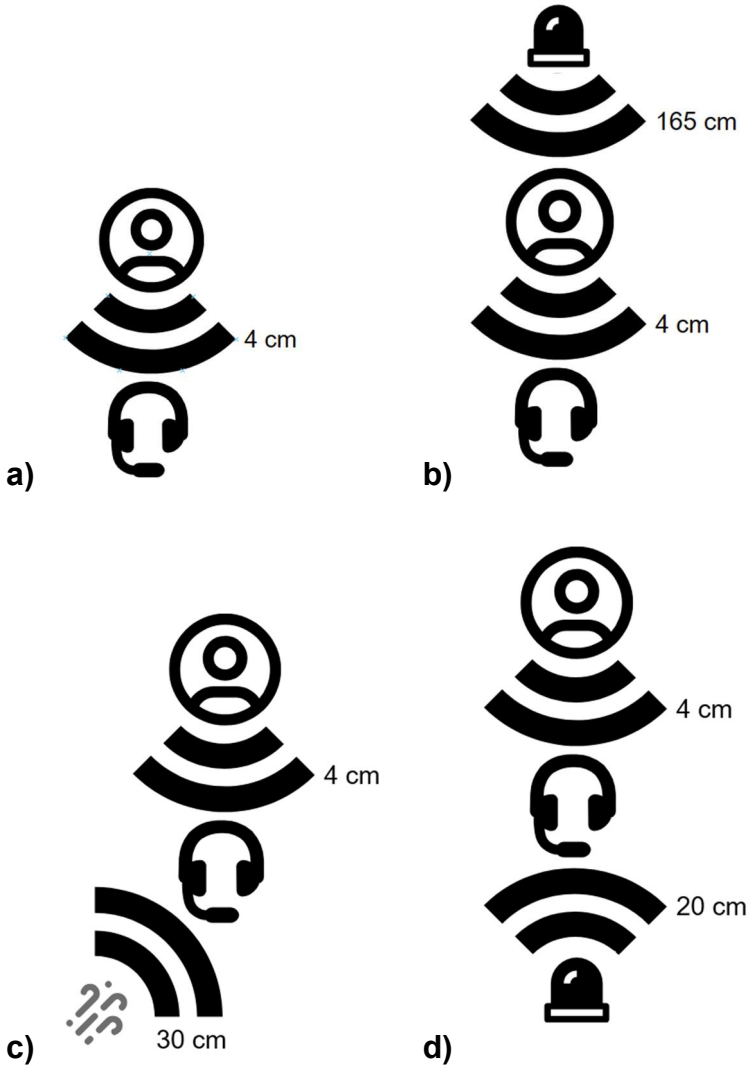
Os testes foram realizados de maneira controlada, buscando reproduzir situações próximas às condições reais de uso. Foram avaliados diferentes cenários de operação, variando características do ambiente e parâmetros de processamento do sistema para analisar o tempo de resposta, consumo computacional e precisão do modelo.

4.1. Acurácia na detecção da *Wake Word*

A avaliação da acurácia do sistema de detecção da palavra de ativação foi realizada considerando diferentes cenários de ruído e três métodos distintos de *downsampling* do sinal de áudio: *nearest neighbor*, interpolação linear e média por blocos. Todos os testes foram conduzidos utilizando o mesmo modelo de *wake word*, mantendo constantes os demais parâmetros do sistema, de forma a isolar o impacto do método de *downsample* no desempenho final.

Os ensaios, ilustrados na Figura 27, foram organizados em quatro cenários: O teste sem ruído foi realizado em ambiente de laboratório, com ar-condicionado ligado e ausência de conversas próximas, representando uma situação controlada de baixo ruído. O cenário de ruído distante buscou simular o ruído do motor da embarcação, que se encontra a aproximadamente 1,65 m do piloto. Devido às limitações físicas do motor que estava em manutenção no período de testes, utilizou-se uma furadeira posicionada à mesma distância como fonte sonora. As rajadas de vento foram simuladas manualmente com o uso de um leque durante a fala, enquanto o teste extremo consistiu na aplicação de vento contínuo gerado por uma furadeira direcionada ao microfone, a aproximadamente 20 cm de distância, em sentido contrário ao da fala.

Figura 27 - Cenários de teste de ativação direta



Fonte: Do autor (2025)
Nota: Representação esquemática. As figuras representam os cenários de teste sem ruído (a), ruído distante (b), rajadas de vento (c) e ruído extremo (d).

Foram realizadas cinquenta tentativas de ativação da *wake word* para cada combinação entre cenário de teste e método de *downsampling* avaliado. Além disso, os testes foram realizados com o microfone equipado com espuma anti-puff.

Além dos testes de ativação direta apresentados acima, foi conduzido um teste adicional de leitura de um texto corrido contendo palavras foneticamente semelhantes à palavra de ativação, com o objetivo de identificar falsas ativações. O texto lido foi o mesmo para os três métodos de *downsampling* e as ocorrências de falsos positivos foram contabilizadas tanto durante este teste específico quanto ao longo dos demais cenários experimentais.

Os resultados obtidos estão apresentados na Tabela 4. Observa-se que o método de interpolação linear apresentou o maior número de falsos positivos, totalizando seis ocorrências, sendo três associadas a rajadas de vento durante os testes de ativação direta e outras três durante a leitura do texto no teste de falsos positivos. Em termos de acurácia, esse método apresentou desempenho inferior aos outros métodos.

Com desempenho intermediário, temos o método de *nearest neighbor*. Embora tenha mantido taxas de acerto satisfatórias nos cenários sem ruído e com ruído distante, foram observados quatro falsos positivos ao longo dos testes.

O método de média por blocos se destacou por apresentar apenas um falso positivo durante todos os testes realizados. Esse método obteve também as maiores taxas de acerto nos cenários sem ruído e com ruído distante, mantendo desempenho semelhante aos demais métodos nos cenários mais severos.

Tabela 4 - Comparativo de acurácia para algoritmos de reamostragem

	Falsos Positivos	**Sem ruído**	**Ruído distante**	**Rajadas de vento**	**Ruído extremo**
Nearest Neighbor	4	88%	82%	56%	46%
Interpolação Linear	6	80%	78%	62%	44%
Média por bloco	**1**	**92%**	**88%**	**58%**	**48%**

Fonte: Do autor (2025)

De forma geral, os resultados indicam que, embora as diferenças de acurácia entre os métodos sejam relativamente pequenas. A média por blocos apresentou o melhor compromisso entre acurácia e quantidade de falsos positivos, reforçando sua adequação para aplicação em um sistema embarcado de reconhecimento de voz em tempo real, operando sob restrições de processamento e em ambientes sujeitos a interferências acústicas.

4.2. Acurácia no reconhecimento de fala

Para avaliar a acurácia do sistema de reconhecimento de fala, foram realizados testes controlados consistindo na emissão de comandos de voz válidos em condições normais de operação, semelhante ao cenário representado na Figura 27b. Foram realizadas 60 tentativas, resultando em 54 comandos corretamente reconhecidos, correspondendo a uma taxa de acerto de 90% e em 6 tentativas o comando não foi reconhecido, não resultando em qualquer ação por parte do sistema. Não foram observados casos de falsa ativação ou execução incorreta de comandos.

Esses resultados demonstram que o sistema apresenta elevada confiabilidade no reconhecimento de comandos válidos, ao mesmo tempo em que mantém um comportamento conservador frente a incertezas, priorizando a não execução de ações indevidas. Essa característica é particularmente relevante no contexto da embarcação, no qual falsas ativações podem representar riscos operacionais.

4.3. Consumo computacional

O consumo computacional do sistema foi avaliado por meio do monitoramento da utilização da CPU durante os testes de ativação da palavra de ativação. Para essa análise, utilizou-se a ferramenta HTOP, que permite observar em tempo real a carga de processamento no sistema durante o funcionamento contínuo do reconhecimento de palavra de ativação.

Os valores apresentados na Tabela 5 referem-se ao período em que o sistema permanece em estado de escuta, aguardando a detecção da palavra de ativação. Nesse estado, o processamento está restrito à captura de áudio, ao pré-processamento do sinal e à execução do modelo de detecção da *wake word*, variando apenas o método de *downsampling* empregado.

Tabela 5 - Utilização de CPU para wake word detection

Método	Mínimo	Máximo
Nearest neighbor	1,2%	2,5%
Interpolação linear	1,3%	2,6%
Média por blocos	1,8%	3,2%

Fonte: Do autor (2025)

Os resultados indicam que, embora seja possível observar diferenças no consumo de CPU entre os métodos avaliados, todas as abordagens apresentam baixa utilização de processamento durante a fase de escuta contínua. O método de média por blocos apresentou consumo superior, o que é esperado em função do maior número de operações envolvidas no cálculo da média de amostras, quando comparado à simples seleção direta de amostras ou à interpolação linear.

Vale destacar que os maiores picos de utilização da CPU não ocorrem durante a etapa de detecção da palavra de ativação. Esses picos são observados somente após a ativação do sistema, momento em que o modelo de reconhecimento de fala baseado na biblioteca Vosk é carregado e passa a processar os comandos de voz do usuário. Dessa forma, o consumo computacional associado ao *downsampling* e à detecção da *wake word* representa apenas uma fração do custo total do sistema durante sua operação completa.

Apesar de não terem sido apresentados valores quantitativos para o consumo de CPU durante a etapa de reconhecimento de comandos de voz pela Vosk, observou-se empiricamente, por meio do monitoramento com a ferramenta HTOP, a ocorrência de picos elevados de utilização logo após a ativação do sistema, atingindo valores próximos a 80% ou 90%, seguidos de uma redução significativa após alguns segundos. Esse comportamento está associado ao carregamento do modelo acústico e linguístico da biblioteca Vosk, bem como à execução inicial do processo de decodificação da fala.

Além disso, o sistema foi implementado de forma predominantemente sequencial, sem o uso de paralelismo explícito ou *multithreading*, o que pode resultar na ocupação intensa de um único núcleo de processamento durante essa fase. Dessa forma, os valores elevados observados não refletem necessariamente uma carga computacional, mas sim um uso concentrado e transitório da CPU, dificultando a obtenção de métricas estáveis e diretamente comparáveis para essa etapa do processamento.

Esses resultados indicam que o impacto do método de *downsampling* no consumo computacional é relativamente pequeno quando comparado às demais etapas do sistema, permitindo que a escolha do método seja guiada principalmente por critérios de acurácia, sem comprometer significativamente a viabilidade. Além disso, apesar destes picos transitórios, a utilização do sistema manteve-se estável,

não sendo observados gargalos significativos decorrentes da implementação proposta.

4.4. Latência na detecção da *Wake Word*

A latência do sistema foi avaliada com o objetivo de quantificar o tempo de resposta entre a pronúncia da palavra de ativação e sua detecção efetiva pelo sistema. Essa métrica é particularmente relevante em aplicações de interação por voz em tempo real, nas quais atrasos excessivos podem comprometer a usabilidade e a percepção de responsividade do sistema.

Devido à natureza acústica da entrada e à ausência de um marcador digital explícito associado ao término da fala humana, adotou-se um método experimental baseado em instrumentação acústica. O sistema foi configurado para reproduzir automaticamente um sinal sonoro imediatamente após a detecção da palavra de ativação, permitindo a marcação temporal do evento de reconhecimento.

Durante os experimentos, o áudio ambiente foi gravado continuamente, contendo tanto a fala do usuário quanto o sinal sonoro de confirmação emitido pelo sistema. A análise posterior da gravação permitiu identificar o instante de término da pronúncia da palavra de ativação e o início da reprodução do sinal sonoro, sendo a diferença entre esses instantes considerada como a latência do sistema.

Observou-se a ocorrência de ativações antecipadas, com tempos inferiores a 100 ms, nas quais o sistema identificou a palavra de ativação antes do término completo de sua pronúncia. Esses casos não foram classificados como falsas ativações, uma vez que indicam a capacidade do modelo de reconhecer padrões parciais da palavra. Contudo, por não representarem o tempo total de inferência após a emissão completa do comando, esses valores foram retirados da análise.

Após a remoção dessas ocorrências, os três métodos de downsampling apresentaram médias de latência semelhantes, conforme apresentado na Tabela 6. Ressalta-se que o tempo medido engloba, na prática, as etapas de bufferização do áudio, *downsampling*, extração de características (MFCC), inferência do modelo, lógica de decisão e reprodução do sinal sonoro utilizado como referência temporal.

Tabela 6 - Comparativo de Latência na detecção da *wake word*

Métrica	Nearest neighbor	Interpolação Linear	Média por blocos
Amostras (N)	25	24	26
Média (s)	0.460	0.463	0.496
Mediana (s)	**0.454**	**0.454**	**0.499**
Desvio Padrão (s)	0.224	0.214	0.205
Mínimo (s)	0.117	0.102	0.151
Máximo (s)	0.814	0.814	0.861

Fonte: Do autor (2025)

Os resultados indicam que os diferentes métodos de *downsampling* apresentam latências médias semelhantes, com variação pouco significativa entre o *nearest neighbor* e a interpolação linear. O método de média por blocos apresentou um aumento um pouco maior na latência média e na mediana, o que é esperado com o maior custo computacional associado ao cálculo da média de múltiplas amostras para cada ponto reamostrado. Ainda assim, os valores observados são compatíveis com uma aplicação interativa, não comprometendo a responsividade do sistema durante a detecção da palavra de ativação. O desvio padrão semelhante entre os métodos indica também que a variação de latência do processamento se manteve estável independentemente da técnica adotada.

4.5. Latência no reconhecimento de fala

O objetivo deste experimento foi avaliar a latência de resposta do sistema de reconhecimento automático de fala, considerando exclusivamente o processamento realizado pela biblioteca Vosk em ambiente embarcado. Para isso, foi medido o intervalo de tempo entre o término acústico da reprodução de um comando de voz válido (“ligar barco”) e o instante em que o sistema indica o envio bem-sucedido da mensagem CAN correspondente ao comando reconhecido.

Como o evento inicial do processo é de natureza analógica, associado à fala humana captada pelo microfone, não é possível medir essa latência de forma puramente digital por meio de *timestamps* internos do software. Dessa forma, adotou-se uma abordagem de instrumentação experimental, na qual o sistema gera um

evento de saída claramente identificável no domínio do tempo, permitindo a análise precisa do atraso entre o fim da fala e a resposta do sistema.

Esse procedimento permite isolar a latência associada às etapas de processamento de áudio, extração de características e inferência do modelo de reconhecimento de fala, desconsiderando o tempo de resposta do módulo MIC, responsável exclusivamente pela execução física da ação na embarcação. Assim, a métrica obtida reflete a latência do sistema de reconhecimento de fala, independente dos atrasos introduzidos pelos atuadores.

Os testes foram realizados utilizando três variações de downsampling previamente mencionadas. Para cada configuração, foram coletadas múltiplas amostras de latência, a partir das quais foram calculados valores médios e o desvio padrão, permitindo avaliar a consistência e a repetibilidade das medições. Os resultados obtidos são apresentados na Tabela 7.

Tabela 7 - Comparativo de Latência no reconhecimento de fala

Métrica	***Nearest Neighbor***	**Interpolação Linear**	**Média por blocos**
Amostras (N)	28	26	23
Média (s)	1.767	1.750	1.857
Mediana (s)	**1.780**	**1.797**	**1.823**
Desvio Padrão (s)	0.239	0.167	0.171
Mínimo (s)	1.308	1.337	1.597
Máximo (s)	2.727	2.013	2.163

Fonte: Do autor (2025)

O método *Nearest Neighbor* apresentou um pico isolado de latência máxima, o que contribuiu para um desvio padrão mais elevado. No entanto, como esse comportamento não se repetiu nas demais medições, o valor foi mantido na análise por representar um cenário válido de pior caso observado durante a operação do sistema.

Os valores de latência observados são razoáveis para a execução de reconhecimento de fala em uma plataforma de desenvolvimento com recursos computacionais limitados. Apesar de não terem sido definidos limites máximos de latência, os resultados indicam que o Vosk oferece desempenho suficiente para aplicações interativas, permitindo que o assistente de voz responda de forma consistente e utilizável no contexto operacional da embarcação.

4.6. Desempenho geral e pontos de melhoria

Durante os testes, observou-se que o uso do Vosk em conjunto com um modelo de comandos restritos apresentou desempenho satisfatório mesmo em hardware limitado. A execução local do reconhecimento de fala eliminou a necessidade de comunicação com serviços externos, reduzindo latência e aumentando a confiabilidade do sistema em ambientes com conectividade limitada ou inexistente.

A arquitetura demonstrou-se adequada para aplicações em tempo real, equilibrando acurácia no reconhecimento, consumo de recursos e confiabilidade operacional, características essenciais para o módulo pensado.

Como pontos de melhoria do projeto, destaca-se inicialmente a substituição da Raspberry Pi 3 por uma Raspberry Pi 4 em versões futuras, uma vez que os testes evidenciaram picos elevados de utilização da CPU durante a execução simultânea do módulo de controle por voz e da interface do sistema. A adoção de um hardware com maior capacidade de processamento contribuiria para maior estabilidade, menor latência no reconhecimento de comandos e melhor experiência de uso.

Além disso, recomenda-se a utilização de uma espuma anti-vento no microfone, visando reduzir a interferência causada pelo vento e por ruídos ambientais na captura do áudio, aspecto especialmente relevante em um contexto de operação embarcada ao ar livre.

A utilização de uma arquitetura *multithread* pode ser considerada um ponto de melhoria do projeto, pois permitiria separar tarefas como captura de áudio, detecção da palavra de ativação, interface e comunicação via rede CAN, reduzindo bloqueios e melhorando a responsividade geral do sistema, contribuindo também para um melhor aproveitamento do tempo de processamento disponível.

Multithreading é uma técnica de programação que permite que um programa execute atividades independentes como threads simultâneos, melhorando a responsividade e a utilização dos recursos do processador (ORACLE, 2020). Seu principal benefício é possibilitar que operações independentes sejam executadas em paralelo, melhorando a responsividade, o aproveitamento do processador e reduzindo bloqueios causados pela execução sequencial de tarefas.

Entretanto, o reconhecimento de fala baseado na biblioteca Vosk é responsável pela maior parte do consumo computacional do sistema e não é thread-safe, o que

impede sua execução paralela em múltiplos threads. Dessa forma, o processamento da Vosk deve permanecer em um único thread dedicado, limitando os ganhos de desempenho que o *multithreading* poderia oferecer. Ainda assim, a separação das demais funcionalidades em threads independentes representa uma melhoria válida, mesmo que não elimine completamente os picos de uso da CPU.

Por fim, destaca-se que o modelo de detecção da palavra de ativação pode ser continuamente aprimorado a partir da coleta de novos dados de áudio e do retreinamento do modelo. Essa flexibilidade permite que o sistema seja ajustado ao longo do tempo para atender a novas especificações da equipe ou para melhorar seu desempenho em diferentes condições de operação.

5. CONCLUSÃO

Este trabalho teve como objetivo desenvolver e validar um assistente por voz embarcado capaz de reconhecer comandos em língua portuguesa e transmiti-los via rede CAN a um módulo de controle da embarcação, visando reduzir a carga de trabalho do piloto em provas de longa duração. Conforme apresentado na introdução, a necessidade de interação constante com os sistemas de controle pode gerar fadiga física e mental, impactando desempenho e segurança. Nesse contexto, a proposta do Módulo de Controle por Voz (MCV25) buscou oferecer uma interface alternativa de interação homem-máquina baseada em fala.

Os resultados obtidos demonstram que o sistema foi capaz de detectar de forma confiável a palavra de ativação e reconhecer comandos previamente definidos, realizando sua interpretação e transmissão estruturada ao módulo MIC19 por meio da rede CAN. A integração entre detecção da palavra de ativação com Edge Impulse, reconhecimento de fala com a biblioteca Vosk e comunicação via rede CAN foi implementada com sucesso em uma Raspberry Pi 3B+, validando a viabilidade técnica da solução mesmo em ambiente sujeito a ruído.

Em termos de desempenho, o sistema apresentou tempos de resposta compatíveis com a aplicação proposta e acurácia satisfatória para o conjunto de comandos definidos, demonstrando que tecnologias de reconhecimento de voz podem ser aplicadas em sistemas embarcados sem dependência de conexão com a internet. Além disso, a arquitetura desenvolvida garantiu que os comandos de voz não interferissem no controle manual prioritário da embarcação, atendendo aos requisitos de segurança estabelecidos nos objetivos específicos.

Entretanto, foram identificadas limitações relacionadas principalmente ao consumo computacional do módulo de reconhecimento de fala e às restrições de hardware da plataforma utilizada. A Raspberry Pi 3B+ opera próxima ao seu limite em cenários de processamento contínuo de áudio, o que pode impactar estabilidade em condições mais severas de ruído ou carga adicional do sistema.

Como sugestões para trabalhos futuros, destacam-se a adoção de hardware com maior capacidade de processamento, como versões mais recentes da Raspberry Pi ou plataformas dedicadas a aplicações de inteligência artificial embarcada, aprimoramentos no sistema de captura de áudio, incluindo microfones direcionais ou técnicas mais avançadas de redução de ruído, expansão do conjunto de comandos e

integração com outros subsistemas da embarcação e realização de testes extensivos em ambiente real de competição para avaliação estatística de taxa de acerto, latência e taxa de falsos positivos.

Dessa forma, conclui-se que o projeto atingiu seu objetivo principal ao demonstrar a viabilidade técnica da implementação de um assistente por voz embarcado totalmente local, integrado à rede CAN e adequado ao contexto do Desafio Barco Solar Brasil. O trabalho estabelece uma base para a evolução da interface por voz na embarcação e criação de novas automações e integrações com outros módulos, contribuindo para o avanço de soluções mais intuitivas, seguras e tecnologicamente integradas na embarcação movida a energia solar.

REFERÊNCIAS

ALPHACEPHEI. **Vosk Speech Recognition Toolkit**. 2022. Disponível em: https://alphacephei.com/Vosk. Acesso em: 11 maio 2025.

ALTHNIAN, A. et al. **Impact of Dataset Size on Classification Performance***:* An Empirical Evaluation in the Medical Domain. Applied Sciences, v. 11, n. 2, p. 796, 15 jan. 2021. DOI: 10.3390/app11020796. Disponível em: https://www.mdpi.com/2076-3417/11/2/796. Acesso em: 8 jan. 2026.

AO, J. et al. **SpeechT5**: Unified-Modal Encoder-Decoder Pre-training for Spoken Language Processing. *arXiv preprint*, arXiv:2110.07205, 2022. Disponível em: https://arxiv.org/abs/2110.07205. Acesso em: 19 fev. 2026.

AWS. **Model Fit: Underfitting vs Overfitting** — Amazon Machine Learning Documentation. Disponível em: https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html. Acesso em: 3 dez. 2025.

BAEVSKI, A. et al. **wav2vec 2.0**: A Framework for Self-Supervised Learning of Speech Representations. NeurIPS, 2020. Disponível em: https://arxiv.org/abs/2006.11477. Acesso em: 19 fev. 2026.

BOSCH. **CAN Specification Version 2.0**. Robert Bosch GmbH, 1991. Disponível em: http://esd.cs.ucr.edu/webres/can20.pdf. Acesso em: 11 maio 2025.

BRANDSTEIN, M.; WARD, D. **Microphone Arrays***: Signal Processing Techniques and Applications*. Berlin; Heidelberg: Springer Science & Business Media, 2001. 398 p. Disponível em: https://books.google.com.br/books?hl=pt-BR&id=nND6ObXSNoEC. Acesso em: 10 jan 2026.

COQUI AI. **Coqui STT***: Open-source speech-to-text engine*. Disponível em: https://stt.readthedocs.io/. Acesso em: 20 fev 2026.

DAVIS, Steven B.; MERMELSTEIN, Paul. **Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences**. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 28, n. 4, p. 357–366, 1980. Disponível em: https://ieeexplore.ieee.org/document/1163420. Acesso em: 15 maio 2025.

DESAFIO SOLAR BRASIL. **O desafio**. Disponível em: https://desafiosolar.com.br/odesafio/. Acesso em: 18 fev. 2026.

EDGE IMPULSE. **Edge Impulse Documentation**. Disponível em: https://docs.edgeimpulse.com/. Acesso em: 17 dez. 2025.

FOSTER, Kelsey. **Top 8 open source STT options for voice applications in 2025**. AssemblyAI, 17 set. 2025. Disponível em: https://www.assemblyai.com/blog/top-open-source-stt-options-for-voice-applications. Acesso em: 17 dez. 2025.

HUGGINS-DAINES, D. et al. ***PocketSphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices***. Proceedings of ICASSP, 2006. Acesso em: 15 maio 2025.

KALDI. **About the Kaldi project**. 2009. Disponível em: https://kaldi-asr.org/doc/about.html. Acesso em: 17 dez. 2025.

KOLUGURI, N. V. et al. **NVIDIA NeMo**: A toolkit for conversational AI. arXiv preprint arXiv:2108.10314, 2021. Disponível em: https://arxiv.org/abs/2108.10314. Acesso em: 19 fev. 2026.

LINUX FOUNDATION. **SocketCAN Documentation**. 2017. Disponível em: https://www.kernel.org/doc/Documentation/networking/can.txt. Acesso em: 11 maio 2025.

LOIZOU, P. C. **Speech Enhancement**: Theory and Practice. 2. ed. Boca Raton: CRC Press, 2013. 711 p. Disponível em: https://www.routledge.com/Speech-Enhancement-Theory-and-Practice-Second-Edition/Loizou/9781466504219

MICROCHIP. **MCP2515 Stand-Alone CAN Controller with SPI Interface**. Datasheet, 2005. Disponível em: https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf. Acesso em: 28 maio 2025.

NATIONAL INSTRUMENTS. **Controller Area Network (CAN) Protocol Overview**. Austin: National Instruments, 2025. Disponível em: https://www.ni.com/en/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/controller-area-network--can--overview.html. Acesso em: 17 dez. 2025.

OPPENHEIM, Alan V.; SCHAFER, Ronald W. **Processamento de Sinais em Tempo Discreto**. 3. ed. São Paulo: Pearson, 2010.

ORACLE. **Multithreaded Programming Guide**: Benefits From Multithreading. Disponível em: https://docs.oracle.com/cd/E37838_01/html/E61057/mtintro-68348.html. Acesso em: 18 dez. 2025.

RABINER, Lawrence R.; JUANG, Biing-Hwang. **Fundamentals of speech recognition**. Englewood Cliffs: Prentice Hall, 1993. Disponível em: https://www-l2ti.univ-paris13.fr/~dauphin/Rabiner93.pdf. Acesso em: 17 maio. 2025.

RADFORD, A. et al. **Robust Speech Recognition via Large-Scale Weak Supervision**. *arXiv preprint*, arXiv:2212.04356, 2022. Disponível em: https://arxiv.org/abs/2212.04356. Acesso em: 19 fev. 2026.

ZÊNITE SOLAR. **Zênite Solar – Embarcação movida a energia solar**. Disponível em: https://zenitesolar.com/. Acesso em: 07 maio 2025.

APÊNDICE A - REPOSITÓRIO DO PROJETO

O código-fonte desenvolvido assim como alguns dos áudios e vídeos utilizados nos testes está disponível publicamente no repositório GitHub abaixo. Alguns dos vídeos eram muitos longos para serem publicados mesmo após compressão, mas é possível ter acesso aos áudios capturados em cada um dos testes antes do *downsampling* e após cada um dos modelos de *downsample* onde Quality 0 é o *nearest neighbor*, Quality 1 é a interpolação linear e Quality 2 é a média por blocos. Cada algoritmo foi divido ainda em *ds* – pasta com os áudios após *downsampling* – e raw – pasta com os áudios antes de qualquer processamento na Raspberry.

Link: https://github.com/ZeniteSolar/MCV25